
Wayne State University Dissertations

1-1-2016

Emerging Genres Of Online Technical Communciation

Luke Anthony Thominet
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_dissertations



Part of the [Other Communication Commons](#), and the [Rhetoric Commons](#)

Recommended Citation

Thominet, Luke Anthony, "Emerging Genres Of Online Technical Communciation" (2016). *Wayne State University Dissertations*. Paper 1597.

This Open Access Dissertation is brought to you for free and open access by DigitalCommons@WayneState. It has been accepted for inclusion in Wayne State University Dissertations by an authorized administrator of DigitalCommons@WayneState.

EMERGING GENRES OF ONLINE TECHNICAL COMMUNICATION

by

LUKE THOMINET

DISSERTATION

Submitted to the Graduate School

of Wayne State University

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2016

MAJOR: ENGLISH (Rhetoric and Composition)

Approved By:

Advisor Date

Date

Date

Date

DEDICATION

I would be nowhere without my family, so I dedicate this work to them. To my father with his sage rock advice. To my brother who is as loyal a friend as any might hope for. To my mother whose patience (and willingness to be an extra set of eyes for my scholarship) never ceases to amaze me. To Allison, who supports me when I need it most. And finally, also to Goob. You are missed. Thank you all for so much support and love over the years.

ACKNOWLEDGEMENTS

I especially want to thank my director, Jeff Pruchnic, for all the long meetings and for helping me make it through when I started to question my project. I also want to thank my committee, Richard Marback, Donnie Sackey, and Samantha Blackmon, for providing me with feedback and supporting my development as a researcher. And I would be remiss to not thank Ellen Barton, who took me under her wing as her research assistant and has continually pushed me toward better research design and better methods.

I also owe a debt of gratitude to the reviewers at SIGDOC and *TCQ*. A section from first body chapter was revised and published in the SIGDOC 2015 proceedings. And my second body chapter was significantly revised and published as an article in the Summer 2016 issue of *TCQ*. The reviewers for both helped push me to make better arguments and to hone my writing.

TABLE OF CONTENTS

Dedication.....	ii
Acknowledgements.....	iii
List of Tables	viii
List of Figures.....	ix
Introduction.....	1
Methodology.....	3
Chapter Descriptions.....	5
Chapter 1 Crowdsourced Documentation Wikis	7
Overview of the Genre.....	8
Value of the Genre	10
Destabilizations in the Documentation Genre System.....	11
Additional Value for Professionals.....	13
Critique of the Genre	14
Analysis of Author support.....	21
Obligatory Content Areas	23
Conventional Content Areas	26
Optional Content Areas.....	27
Critique of Current Author Support.....	31
Choose an Action-Oriented Approach.....	34
Anchor the Tool in the Task Domain	35
Support Error Recognition and Recovery.....	36
Support Reading to Do, Study and Locate.....	37

Conclusion	37
Chapter 2 Player Video Game Reviews	39
Methods.....	40
Development of Coding System.....	41
Sample.....	46
Desura	47
Gamespot	47
Giant Bomb.....	48
GOG (Good Old Games)	48
Metacritic	48
Steam.....	49
Basic Differences Between Revenue Models	49
Study Limitations.....	50
Results.....	50
Discussion.....	53
Background Information Move.....	54
Specific Aspects Move	57
Overall Quality Move	60
Narrate Game Experience Move.....	63
Offer Advice Move	64
Influence of Individual Games on Reviews.....	66
Summary of Variation.....	68
Influences on a Changing Genre.....	69

Conclusion	72
Future Research	74
Chapter 3 Open Development of Video Games	75
Activity Theory Analysis	77
The Activity of Open Development.....	79
OD Activities	86
OD Genres	99
Chats	101
Comments	102
Dev Sites	102
Digital Distribution Store Pages	104
Fan Art	105
Forums	105
Game Design Documents	107
Game Iterations	108
In-Game Feedback	108
Player-Built Content	109
Production Trackers	109
Reviews (and Previews).....	110
Social Media Posts	111
Update Announcements	112
Videos	113
Votes	115

OD Genres Summary	115
Feedback Activity and Genre Knowledge	117
Conclusion: Pedagogy of User-Generated Genres	122
Crowdsourced Documentation Wikis	124
Player Game Reviews	125
Open Development	126
Appendix A: Review Descriptors	129
Appendix B: Frequency of moves and strategies by revenue type.....	135
Appendix C: Frequency of moves and strategies by site	136
References	137
Abstract	164
Autobiographical Statement	166

LIST OF TABLES

Table 1.1: Analysis of Number of Distinct Authors for Documentation Wikis	17
Table 1.2: Analysis of Number of Edits for Documentation Wikis	17
Table 1.3: Analysis of Number of Major Edits for Documentation Wikis.....	19
Table 1.4: Content Areas of Author Support Guides.....	22
Table 2.1: Existing User Review Coding Systems	43
Table 2.2: Moves in Video Game Player Review Genre.....	45
Table 2.3: Overview of Websites in Sample	47
Table 2.4: Overall Frequency of Moves	50
Table 2.5: Frequency of Moves as Opening and Closing of Reviews.....	51
Table 2.6: Review 92 Separated by Moves	52
Table 2.7: Comparison of Move Frequency by Site Revenue Type.....	53
Table 2.8: Comparison of Move Frequency by Website	53
Table 2.9: Description of Background Information Strategies.....	55
Table 2.10: Description of Specific Aspects Strategies.....	59
Table 2.11: Reviews without Overall Quality Move	61
Table 2.12: Description of Overall Quality Strategies.....	62
Table 2.13: Steam Reviews' Use of Summarize Quality in Rating Strategy	63
Table 2.14: Description of Overall Quality Strategies.....	65
Table 2.15: Opening Moves for Reviews of Binding of Isaac on Giant Bomb.....	67

LIST OF FIGURES

Figure 3.1: Activity Theory Diagram	78
Figure 3.2: Open Development as a Unified Activity	84
Figure 3.3: Two Interacting Activity Systems	86
Figure 3.4: Open Development as Design Communication Activity	89
Figure 3.5: Open Development as a Marketing Activity	92
Figure 3.6: Open Development as Feedback Activity	96
Figure 3.7: Combined O.D. Activity	97
Figure 3.8: Genres of O.D. Activities	117

INTRODUCTION

As social web technologies continue to increase individuals' access to distributed audiences, technical communication practices must shift to address their emerging roles in networked communication. Whereas, in the past, the definition of technical communication was limited to the communication practices of experts working within organizations, more recent scholarship has expanded the definition through a recognition of 1) the participation by non-experts in areas like online instructions and social media (Verzosa Hurley & Kimme Hea, 2013; Zachry & Ferro, 2014; Johnson-Eilola & Selber, 2008) and 2) the continually changing networks of workers and work activities as systems of post-Fordist capitalism have forced organizations into more fluid forms (Spinuzzi, 2007; Hart-Davidson, 2012). These changes emphasize the importance of expanding research on User-Generated Technical Communication (UGTC): a set of emerging practices characterized by loosely organized communities of non-professionals producing technical communication.

Given the rapidly shifting practices that characterize most UGTC work, I use a broad definition of UGTC throughout my dissertation. First, I draw on STC's (n.d.) definition of technical communication as "communicating about technical or specialized topics, such as computer applications." In this way, my chapters explicitly address three key discussions around software technologies: use support, reception, and development. Likewise, I draw on Daugherty, Eastin, and Bright's (2008) definition of user-generated content (UGC) as "media content created or produced by the general public rather than by paid professionals and primarily distributed on the Internet." However, even this broad definition is problematic given the numerous rearticulations of similar concepts under different terminology (see, for example, Bowman and Willis' (2003) discussion of participatory journalism, Powazek's (2006) coining of "authentic

media,” the ongoing discussion of prosumption (Ritzer & Jurgenson, 2010), and even Bruns’ (2008) rearticulation of the latter concept as “produsage”). For the sake of the discussion of UGTC, the above definition of UGC had one major oversight: in technical communication scholarship, we’ve generally come to accept a range of theories (activity, actor-network, ecological rhetoric, etc.) that all see writing as situationally influenced (or even as spread evenly through a number of largely undifferentiated agents). Even a relatively limited version of these views could not ignore the situational constraints on the authoring of UGC, most notably the platforms it is authored with and the websites it is published on—all of which are, almost universally, not created by users. Thus it becomes necessary, even while studying UGTC, to also consider the influence of professional technical communicators on this content. Ultimately then, this study of UGTC is situated at the intersection between amateur and professional communicators and thus has implications for the future communication practices of both.

My dissertation specifically explores three emerging practices of UGTC: crowdsourced wiki documentation, player game reviews, and video game open development. In the first study, I analyze five crowdsourced documentation wikis and find systemic inconsistency in the workflow and content quality of the documentation. I argue that practitioners should use minimalist documentation theory to design more effective user-centered author support for the wikis. My second chapter uses Bhatian move-strategy analysis to investigate variation in the genre structure of a corpus of 180 video game reviews sampled from six websites. Based on the results, I argue that emerging genre variations respond to the exigencies of specific sites and also to new types of audiences and purposes. My final body chapter explores communication practices tied to video game open development, a new methodology where game prototypes are publically distributed in order to support transparency. By tracing the activities and genres used to facilitate this

methodology, I argue that breakdowns in open development activity can be associated with differing goals among stakeholder communities and that a deficit in genre knowledge lowers the usefulness of some feedback communications. Overall, my dissertation explores these three practices as a means to conceptualize the types of technical communication work that users are engaging in and to consider how that work might be changed for the better.

METHODOLOGY

Broadly speaking, my dissertation uses a rhetorical genre studies approach inspired by Russell's (1997) integration of genre and cultural-historical activity theory and Spinuzzi's (2003) genre tracing methodology that integrated macro, meso, and microscopic levels of research and analysis while focusing on genre ecologies. By using this methodology, which is founded in activity theory, and its associated methods, I am able to examine the emerging genres without reducing them.

Genre has a long history in technical communication scholarship. The traditional prescriptive approach described genres as a set of fixed document structures or as something like a checklist of qualities for an ideal document. This concept of genre has been an aspect of the field since its infancy: while most early textbooks concentrated on correctness in style, some, like *A Manual of Engineering Specifications and Contracts* (Haupt, 1881), focused specifically on instilling proper documentation forms (Patricia Sullivan, 2012, p. 226). Likewise modern pedagogies have also drawn from the formal conception of genre: for example, most technical communication textbooks continue to include numerous sections enumerating features of a genre. For example, Markel's (2012) widely used textbook included chapters on correspondence, job application materials, proposals, and lab reports among other genres. Notably, the chapter on lab reports spent only two pages on the process of writing successful lab reports (Markel, 2012, pp.

490–92), but it used six pages on “Understanding the Structure of the Lab Report” (pp. 493-499).¹

Current scholarship, however, has shifted towards a more rhetorical understanding of the role of genre. First, Miller (1984) introduced socio-cultural theories of genre, which argued that real genres were built through local, contextual action. Then Schryer (1993) stressed that even these genres were not static, but were only “stabilized for now” (p. 200). Together these theories say that genres are created in response to repeated, local situations, and they change only through direct action by individuals or organizations. Other scholars have shown how genres evolve in connected systems, ecologies, or networks, and how they can even mediate the range of actions available to a technical communicator (Moeller & Christensen, 2009; Orlikowski & Yates, 1994; Spinuzzi, 2003; Spinuzzi & Zachry, 2000; Yates & Orlikowski, 2002). At the same time, researchers have also emphasized that genre knowledge is localized (Berkenkotter & Huckin, 1995) often within individual organizations (Yates & Orlikowski, 1992). Finally, recent studies have begun to explore the multimodality of genred interaction (Prior, 2009) and the formation of emergent digital genres (Amidon, 2005; Bergquist, Ihlström, Ljungberg, & Åkesson, 2008; Shepherd & Watters, 1998). Together, this research gives us a dynamic understanding of how we come to build genre expectations for texts as we construct and use them.

My dissertation uses this rhetorical genre theory as the basis for genre analysis. I use a range of established methods in analyzing the genres in each chapter. In the first chapter, I primarily use a multiple/mixed methods approach largely inspired by Graham et al.’s (2015) statistical genre analysis methodology. My second chapter uses a genre move-step coding system inspired by Swales (1990), Bhatia (1993, 2002), and Biber, Connor, and Upton (2007). And my third chapter uses activity mapping (Engeström, 2001; Kain & Wardle, 2005; Kaptelinin, n.d.) to

¹ To be fair, Markel (2012) did acknowledge the situated nature of genre knowledge, but he offered an even stronger rejoinder to say that the genre structure was effectively universal (p. 481).

approach genre-mapping methods discussed by Spinuzzi and Zachy (2000), Spinuzzi (2002) and Moeller and Christensen (2009). This range of approaches allows me to match my method to the object of study while maintaining a tight focus on the textual evidence.

CHAPTER DESCRIPTIONS

In Chapter 1, I explore the expansion of crowdsourced documentation wiki. I begin by defining the genre according to the constituent terms (crowdsource, user documentation, and wiki) and discuss its potential value both for the field's scholarship and for its professional practice. Next, I analyze the authoring and editing trends by using data from the wiki history pages. Based on this analysis, I critique the inconsistencies found on the sites and suggest the need study the author support guides on these sites. Finally, after an analysis of these guides, I argue that professionals constructing these sites need to use user-centered design theories to better understand and support the work of amateur authors.

In Chapter 2, I study variations in the player game review genre. First, I establish the relevance of this genre to the field of technical communication. Then I describe the development of my coding system and sample for the study. Based on this method, I discuss both the general genre structure of reviews as well as the localized variations on different sites. Finally, I argue that some of these variations reflect a drift in fundamental genre characteristics (such as purpose or implied audience) and that professionals seeking to support user-generated technical communication communities need to adopt discourse analysis methods to better support the goals of the users.

In Chapter 3, I examine communication surrounding video game open development in order to better understand how technical communication can contribute to this emerging methodology. I open by roughly defining of open development in order to clarify the discussion.

Then I explore how activity theory mapping methods can help us to capture the complexities of the work. Using this method, I iteratively map the multiple activities that, together, comprise open development. Finally, I use these activities as a structure to map the genres used in open development. Ultimately, I argue that this mapping system helps identify breakdowns in open development work and that professional communicators need to build users' genre knowledge in order to realign player goals to more directly contribute to developer goals.

Finally, my conclusion applies the previous chapters' discussions to reconceptualizing technical communication pedagogy. I open this chapter by discussing the changing approach to genre pedagogies in technical communication classrooms. Then I outline both traditional content-based and emergent pedagogical approaches that respond to the genres discussed in each chapter. Finally, I conclude by summarizing the potential impact of teaching writing studies content and approaches in preparing the next generation of technical communicators.

CHAPTER 1 CROWDSOURCED DOCUMENTATION WIKIS

The crowdsourced documentation wiki is an ideal place to start our study of emerging genres of online technical communication. First, the genre's break from traditional methods of production is significant on both practical and theoretical levels. In the new genre, the content is developed, organized, and reviewed by end users, with professional technical communicators playing only a periphery role. At the same time, this genre shows us ways to move closer to user-centered design and distributed work. Second, this genre also ties the field's past to its future. It refigures one of the most central and complex genres of technical communication (user documentation) while also creating environments where professionals and users need to discover new means of collaboration.

Despite this value, crowdsourced documentation wikis also suffer from systemic inconsistency—a flaw that might be acceptable in some contexts, but one that significantly damages the usefulness and usability of any genre of documentation. I will discuss this inconsistency later in this chapter in my analysis of the authoring and editing trends of 150 pages from 5 crowdsourced documentation wikis.

Given its potential value and flawed state, the crowdsourced documentation genre helps us explore how professionals can intercede in user-generated content to create environments conducive to effective technical communication. The latter half of this chapter takes up this discussion by analyzing the current state of author support guides on these wikis and offering some recommendations for improving these guides.

Ultimately, this chapter argues that these author support guides are not effectively designed to support the distributed, inexperienced workers who are contributing to crowdsourced documentation wikis. Consequently, I argue that we can reframe these guides as user

documentation, specifically documentation for the technology of technical writing targeted towards the audience of non-expert writers and for the contextual work of producing user documentation on a wiki, in order to find design theories that better support the needs of this audience.

In the sections that follow, I will define the genre of crowdsourced documentation wikis and establish its value through the literature. Then I will explore the issues inherent to the genre. I will specifically focus on showing how the current genre shows significant and problematic inconsistencies. From there, I will explore the current state of author support on these wikis, focusing not only on the common features of this section of the sites but also on the less common, but important features. Finally, I will describe a potential solution to the problem: the use of minimalist documentation design theory to create guides tailored to the audiences using them.

OVERVIEW OF THE GENRE

Crowdsourced documentation wikis are one part of a growing ecology of social help systems for computer software (and other technologies) that have received significant scholarly attention in recent years (see Berglund & Priestley, 2001; Frith, 2014; Lanier, 2011; Selber, 2010; Swarts, 2015). The crowdsourced documentation wiki genre is closely related to several other genres (including other instructional wikis and non-crowdsourced documentation websites), which necessitates a clear definition of the genre. To create this definition, I drew on scholarly and professional discussions of each of the constituent terms of the genre's name: crowdsource, documentation, and wiki.

In their survey of the literature, Estellés-Arolas and González-Ladrón-de-Guevara (2012) found a considerable level of disagreement on the exact meaning of “crowdsourcing” (p. 189). They integrated the divergent scholarship on the topic into a single definition that clearly identified

the initiator (an individual or organization), the makeup of the crowd (“individuals of varying knowledge, heterogeneity, and number” (Estellés-Arolas & González-Ladrón-de-Guevara, 2012, p. 197), and the voluntary nature of the task.

Likewise, there has been some disagreement on the precise definition of user documentation. While the genre was seen as a varied set of texts (Weiss, 1991, p. 4; Zachry, 1999, p. 23), including both online and print media (Barker, 2002, p. xxii), there was some disagreement over exactly how these texts functioned. Some scholarship presented documentation as supporting users’ pre-existing tasks by helping them use technologies efficiently (Barker, 2002, p. xxii; Weiss, 1991, p. 4). Other scholarship, however, has conceptualized the work of the genre in a more dynamic fashion: “computer documentation ... carries with it the potential to mediate between human activities and computer processes” (Zachry, 1999, p. 23; see also Mirel, 1998). There was value in both of these conceptualizations: the former offered a more direct means to test documentation for effectiveness while the latter encouraged consideration of the complex functions of documentation.

Unlike the other two terms, there was general agreement on which technologies qualified as wikis. While many wiki platforms have been developed (ranging from public software such as MediaWiki to proprietary systems such as PbWorks or Confluence), they have all focused on providing collaborative spaces where users could quickly and easily edit a text (often through wiki markup or an in-line text editor). Most wiki platforms have also included additional features to help communities of authors collaborate such as talk and history pages (see Gentle, 2012, p. 95; Maddox, 2012, p. 16).

Based on these definitions of the constituent terms, I developed the following definition of the crowdsourced documentation wiki genre:

The purpose of a crowdsourced documentation wiki is to support and mediate the use of a technology. It is collaboratively and voluntarily authored by a heterogeneous group of distributed individuals in response to a task posed by an individual or organization. It is written and published using a wiki platform, which allows for quick online editing and also includes typical wiki features.

This definition worked to differentiate the crowdsourced documentation wiki from other genres. For example, crowdsourced documentation has been published using tools other than wikis. This may have changed the authoring and/or use of the genre: an author engaging with a wiki page for the first time could look back over the history of that page to see how it evolved while other software platforms might have obscured this kind of information and caused cyclical edits or problems with version control. Additionally, documentation users might have approached a wiki page differently due to experience with existing wiki sites that encouraged contributions to a knowledge base. In similar ways, the definition could be used to create differentiations based on the crowdsourced and documentation aspects of the genre.

VALUE OF THE GENRE

With a clearer concept of exactly what a crowdsourced documentation wiki is, we can explore the potential value of this genre in more depth. Primarily, this exploration works to further justify the topic of this chapter, but it also offers some insight into how technical communication practice is changing. My argument for value of the crowdsourced documentation wiki genre is two-fold. First, it is valuable both to academics and professionals as a genre that repairs long-standing destabilizations in documentation systems. Second, it offers practical value to professional technical communicators as a means to represent their organization while also

building a relationship with users. I will address each of these arguments in more depth below.

DESTABILIZATIONS IN THE DOCUMENTATION GENRE SYSTEM

Spinuzzi (2003) theorized that genres change because of destabilizations at multiple levels (i.e., contradiction in sociocultural activity, discoordination in goal-directed action, and breakdown in routine, unconscious operations) (p. 30). The creation of crowdsourced documentation wikis (and the larger move toward social documentation in general) can be seen as a response to this kind of systemic destabilization.

The goal of documentation is to meet users where they are and to seamlessly mediate their technology use. Yet most traditional texts on creating documentation have lamented the impossibility of ever fully understanding and supporting the documentation user. For instance, Low (1994) enumerated many ways in which users can become irritated with documentation that does not directly answer their questions (pp. 6-7). Weiss (1991) attributed the difficulty of producing effective documentation to writers knowing too much about a system and subsequently having difficulty “[making] themselves clear to less knowledgeable readers” (p. 7). Johnson (2012) expressed a similar idea to explain why he adopted a documentation wiki system:

It would always surprise me that no matter how well I described tasks and processes in the documentation, users had questions. They used the application in ways I didn’t fully anticipate. Terms confused them. Step sequences were not easy to follow. They couldn’t find information. Some tasks needed to be more visible. Who could anticipate all of this beforehand? (pp. 3-4)

Gentle (2012) described similar reasons for moving towards wikis for documentation, including the limits of any single person’s knowledge and the need to interact with customers (pp. 93-94).

These authors were describing a destabilization within the documentation genre. There was a contradiction in the activity system, where the documentation producers’ domain knowledge (in technical communication and/or system expertise) was failing to meet the needs of community.

At the same time, it was also a discoordination in the goal-directed action of the genre. In order to plan documentation, technical communicators were using a corpus of user-centered design methods but were still finding that end products failed to anticipate users' needs. (This argument should not be construed a full-scale rejection of user-centered design methods, but rather, simply, as a recognition that they imperfect tools.) Crowdsourced documentation wikis have offered one way to correct this destabilization: by leveling the producers and users of the content, these new forms of documentation have been able to organically respond to users' needs.²

We might extend the academic argument for the value of the genre one step further. Not only do crowdsourced documentation wikis work to fix a traditional destabilization in the genre, but, in doing so, they also move toward theories of practice that are highly valued by the field. For example, crowdsourced documentation wikis reflect a movement towards conceptualizing technical communication as symbolic-analytic work. Johnson-Eilola (1996) criticized the way that technical communication in general and documentation in particular were seen by companies as secondary, sometimes superfluous aspects of technology development, and he argued that technical communicators should redefine themselves as symbolic-analytic workers, who “possess the abilities to identify, rearrange, circulate, abstract and broker information” (p. 255). This argument has had a strong influence since its publication, as technical communication scholars continue to search for ways to move practice and pedagogy towards symbolic-analytic work (see, for example, Lippincott, 2003; Pigg, 2013; or Wilson, 2001). Johnson-Eilola (1996), referencing the work of Robert Reich, said that the education of symbolic-analytic workers must include: “collaboration, experimentation, abstraction, and system” (p. 258). Documentation wiki work has

² Of course, we can and will notice further destabilizations within the emerging genre of crowdsourced documentation wikis, but as Schryer (1993) argued, genres are only stabilized-for-now, suggesting that new destabilizations inevitably arise that will result in further genre change.

necessitated a significant focus on these tasks, including: structuring the site, managing systems for collaboration, negotiating knowledge, and designing an effective environment for technical discourse. Documentation wikis has also acted as excellent sites for experimentation with methods and techniques as a range of contributors attempted nontraditional methods and rapidly received responses to their work. At the same time, the move towards crowdsourced documentation also connects with Spinuzzi's (2007) discussion of technical communication as distributed work as "coordinative, polycontextual, crossdisciplinary work that splices together divergent work activities (separated by time, space, organizations, and objectives)" (p. 266). Workers on crowdsourced documentation wikis have had to collaborate with contributors from around the world while bringing together a variety of disciplinary views. The work has been produced and revised over long periods of time in many different locations by a crowd comprised of individuals across organizations. Put together, the fit of these theories with crowdsourced documentation wikis has opened the potential of this genre as a site for further exploration of the future of the field.

ADDITIONAL VALUE FOR PROFESSIONALS

While the primary purpose of documentation is to mediate technology for users, it can also hold value for organizations as a means of representing themselves. Maddox (2012) argued that "documentation is the face of the company. It contributes to the reputation and perceived character of the product and of the organization" (p. 5). In this way, crowdsourcing documentation could help organizations depict themselves as interested in the needs of the users and open to feedback (in much the same way that open source software development has changed public perceptions of those organizations).

Part of this value of representing the company effectively has also come from the

organizations' vested interest in having some control over public discourse on their products. Maddox (2012) discussed this issue while justifying a turn toward social documentation: "What if a reader finds a less authoritative source before they find ours, with perhaps even incorrect information about our products?" (p. 5). Crowdsourced documentation has been and will continue to be spontaneously created by users online, whether it is in the form of blogs, forums, or even wikis, particularly when a technology's ethos (i.e. open source technologies) encourages collaborative contributions. By creating a crowdsourced documentation resource of any kind, organizations could potentially gain some control over the content accuracy of that resource while also limiting the creation of new resources. Given the current capabilities of online discourse, if organizations do not build it themselves, others very well might. Certainly, this would not happen in every single instance, but it could (and does) happen often enough that the activity demands attention.

CRITIQUE OF THE GENRE

Despite the importance of this genre, there have been some significant limitations to its effectiveness. Several authors have already discussed difficulties they experienced while implementing the genre, such as:

- Relatively small groups of contributors (Gentle, 2012, p. 114).
- The high management workload of such sites (Johnson, 2012, p. 40).
- The difficulty of reusing and/or translating wiki content (Gentle, 2012, p. 96; Johnson, 2012, pp. 22–23; Maddox, 2012, p. 219).

While these issues were noteworthy, they did not necessarily undermine the essential value of the new genre. Each might have arisen in specific instances of the genre without necessarily compromising the quality of the documentation.

The inconsistency of the genre, however, has posed a more serious limitation. Crowdsourced documentation wikis need to fulfill the goals of documentation, and consistency is essential for building usability and trust in documentation (see Albers, 2000; Byrne, 2005; Mehlenbacher, 2003). Without consistency, one page may be usable, task-oriented, and aesthetically pleasing, while another is confusing and inaccurate. And, by most measures, crowdsourced documentation wikis have been highly inconsistent. This might be expected, given the range groups of authors contributing to the sites and the minimal oversight. We might even expect the quality of documentation wiki pages to vary in the ways similar to Wikipedia articles³, but the extent of this similarity has not yet been established.

To study this inconsistency, I analyzed the editing and authoring trends on 150 pages from five crowdsourced documentation wikis: The Apache OpenOffice Documentation Project Wiki, The Blender Wiki, The Fedora Documentation Project Wiki, The TinyOS Documentation Wiki, and The Ubuntu Community Help Wiki. I selected these wikis by searching discussions of crowdsourced documentation for references to organizations that were actively engaging in the practice. Two of these sites (Blender and TinyOS) were the official documentation for the software. But these two still took different approaches, with Blender adopting a more linear, manual style, while TinyOS used a less structured format. The other three wikis were not the official documentation for the software. Two of them (OpenOffice and Ubuntu) acted as important supplemental documentation that expanded beyond the official documentation. The OpenOffice wiki was organized into different categories of documentation type (such as how-tos or tutorials) and provided links to resources outside of the wiki (such as tutorials on external blog sites). The

³ However, this argument is also less than straightforward since various studies are not in agreement over Wikipedia's reliability: some have found Wikipedia to be as reliable as other encyclopedias (Giles, 2005), while others found that it is far less reliable than other reference sources (Holman Rector, 2008).

Ubuntu wiki was much more insular and was organized based on the tasks (such as installation or updating) and parts of the system (such as peripherals). Finally, the Fedora wiki did not act as a major part of the organization's documentation. Since the official documentation for Fedora was crowdsourced and published through DocBook, the wiki largely acted as a holding site for supplemental or outdated pieces of documentation, though it did also maintain several pieces of unique and frequently updated documentation.

To analyze these wikis, I first selected sample pages. I generally did so blindly and randomly. In some cases, this was facilitated by a "random article" link on the wikis. In other cases, I selected pages from several sections (or categories) on the wiki. In all cases, I focused solely on documentation pages (as opposed to author support pages and user pages). Overall, I selected thirty pages from each wiki for a total of 150 pages across the five sites.

To explore the consistency of the pages, I initially analyzed two data points (both discoverable on the history page): the total number of edits and the total number of distinct authors. However, the number of edits as recorded on the history page did not accurately represent the real number of edits because authors used different editing methods. Some authors edited the page directly on the wiki and saved each time they made an edit, which resulted in a series of small edits spaced out every few minutes. Other authors apparently worked on the material off of the site and uploaded all of their changes at once, which resulted in fewer and larger edits. To normalize these two disparate editing techniques, I counted any two edits made by the same author within a three hour timespan as a single edit.⁴ Using these two data points, I calculated the mean number of edits and the mean number of distinct contributing authors for pages on each site. Then, to test

⁴ I decided on this 3 hour time span as a common denominator by reading and rereading many of the history pages and looking for groupings of edits. This time span gave the most inclusive groupings while still acknowledging clear divisions.

consistency, I calculated the coefficient of variation (CV) for both numbers.⁵ The results can be found in Tables 1.1 and 1.2 below.

Table 1.1

Analysis of Number of Distinct Authors for Documentation Wikis

Wiki Site	Mean Number of Authors	Coefficient of Variation
Apache Open Office	2.73	0.81
Blender	6.9	0.77
Fedora	9.77	1.81
TinyOS	3.03	0.93
Ubuntu	11.9	1.27
Overall	6.87	1.64

Table 1.2

Analysis of Number of Edits for Documentation Wikis

Wiki Site	Mean Number of Edits	Coefficient of Variation
Apache Open Office	6.3	1.17
Blender	10.33	0.65
Fedora	17.77	2.22
TinyOS	5.57	1.01
Ubuntu	15.9	1.16
Overall	11.17	1.83 ⁶

These results showed that pages on individual sites were inconsistent. For example, we can start by looking at the smallest CV: the number of edits on a Blender wiki page. According to the results, an average page had been edited approximately 10 times. The CV for this statistic was .65, which means that the standard deviation was approximately 6.5 (or nearly two thirds of the mean). In other words, when looking at a random page on this site, readers could assume that it had been edited anywhere from four to sixteen times. The actual numbers from the sample were

⁵ A coefficient of variation is the ratio of the standard deviation in a sample to the mean. Typically, it is used to see if a mean is representative of the sample. A data group with a mean of 5 and a standard deviation of 2 would have a CV of 2/5 or .4. A high CV means that the standard deviation is large in comparison to the mean, suggesting that there is little consistency in the sample.

⁶ Admittedly the “Overall” numbers at the end of each table are fairly meaningless. Given the lack of established standards for the genre, they show the expected result: (that there isn’t a representative way to discuss the number of authors or edits across all documentation wikis).

even more spread: one Blender wiki page had been edited only twice (both times by the same person) while another page was edited 32 times. And this occurred on the most consistent of the five sites (according to both measures described above).⁷ The results were even more varied when looking at the least consistent of the sites, the Fedora Wiki.⁸ The standard deviations here were nearly twice the mean largely because of a single outlier page that received 216 edits from 57 distinct authors. After removing the page from the sample, the means were closer to the other sites: 6.86 authors (1.14 CV) and 10.9 edits (1.16 CV). Still, even this “more consistent” sample involved a standard range of approximately 1-14 authors and 1-20 edits, which demonstrated that consistency in this genre could only be discussed in relative terms. All of the sites were highly inconsistent.

The inconsistency on these pages might have been acceptable if there was an assurance that each page had significant attention and revision after its initial posting. To investigate whether any such work was being done, I calculated the number of major edits that each page received. To do this, I categorized edits by the amount of data changed. I defined major edits as those that added or subtracted 500 or more bytes of data (a similar classification was used by the history pages themselves which bolded the information for any edit changing 500 bytes or more). The results are shown in Table 1.3 below.

⁷ We can attribute this consistency to the policies of the site, on which users were not allowed to add new pages (such decisions were made by a leadership committee). This practice limits the number of single-author pages.

⁸ It should be noted, that, once again, the inconsistency of the Fedora Wiki could be attributed to the management of the site. Since the Fedora Documentation Project saw the DocBook site they run as the primary location for hosting finished (and crowdsourced) documentation, the wiki acted primarily as a testing area for incomplete documentation. This meant that there are few controls on the editing of the site and that many pages were never intended to be updated.

Table 1.3

Analysis of Number of Major Edits for Documentation Wikis

Wiki Site	Mean Number of Major Edits ⁹
Apache Open Office	0.53
Blender	1.07
Fedora	3
TinyOS	0.73
Ubuntu	1.73

On every site, at least 30% (and up to 70%) of the pages had no major edit after the original posting. Every site also had pages with more than six major edits. The low means here were problematic, but the inconsistency itself only reinforced the issues found in the previous analysis.

As an aside, two factors might have complicated these results. First, particularly important or central pages may have received more major edits than periphery or details-related pages. Of course, it is difficult to say definitively which pages were actually “central” to the documentation, but the pattern did seem to have some anecdotal validity: installation pages and getting started pages often had more major edits than the average for the site. Second, major edits could also have accumulated over time, with older pages having more major edits. This explanation also had some validity as there was, on most sites, a weak correlation between a page’s age and the number of major edits it received. However, neither of these explanations showed significant relationships that solved or even lessened the systemic inconsistency across the documentation.

Consistency not only acted as a means for users of documentation to accurately understand how to approach the genre, but also as a crucial form of support for authors as well because documentation is difficult to write, and creating effective documentation is even harder. This was

⁹ I chose not to use the coefficient of variation on this measure because the number of major edits was too low and because the results incremented only on full integrals (i.e., on most sites, the CV would have been high even if there standard deviation was only a single major edit.)

one of Johnson's (2012) major complaints about using a wiki for documentation: "Technical writing usually follows a meticulously detailed, step-by-step procedural approach. You likely have a style guide and general methodology that you follow. Volunteers who try to jump into technical writing tasks usually don't have this background" (p. 37). Consistency in authoring and editing trends is a sign that there is an established workflow and that community practice supports the work of the authors.

Without reasonably consistent authoring and editing trends, we can expect to find some significant issues in the quality of the documentation content. These issues did indeed manifest across the sites used in this analysis. For example:

- The Blender wiki included an incomplete reference section with only 7 items. All of these items were part of the original posting of the page on December 21, 2011. The page remained this way until December 2014 when the wiki was abandoned for a more controlled form of crowdsourced documentation.
- Many pages had an unprofessional or confusing writing style. For example, "This is how to create the virtual machine, what settings I recommend, and why, etc.. For instance, you only need a 4 GB harddrive. Additional space will be put on additional virtual harddrives, but for now, only create a 4 GB harddrive, allocate all space immediately, and do not split into 2GB chunks." ("DinkelServerBase," n.d.)
- Finally, other pages had broken images and links. For example, the "Boomerang External Antena" page on the TinyOS wiki had a broken image. This image had been uploaded in 2008 and was marked as an upload error at that time. However, it still had not been fixed in 2014.

This is not to say that the sites were wholly ineffective, but rather that they were not universally

reliable or effective. Some pages were excellent, others less so.

ANALYSIS OF AUTHOR SUPPORT

In traditional documentation projects inconsistency has typically been addressed through the implementation of style guides (Allen, 1996; Byrne, 2005; Mackay, 1997). And, true to form, most of these documentation wikis had a style guide of their own. However, these guides had to cover a range of topics that lay outside of the traditional purview of organizational or disciplinary guides, so I have chosen to refer to them with the more inclusive term of “author support guides.”

Since professional communicators were already in charge of developing these style guides for most documentation projects, author support guides represent one of the best means for professional technical communicators to effect the consistency of crowdsourced documentation wikis. Several authors have already briefly discussed the importance of developing a guide as part of any such wiki. Gentle (2012) stated that style guides could be useful and referenced several existing guides, but she also said it might be difficult to enforce style guidelines (p. 99). Maddox (2012), likewise, recommended a "light-weight guide that is a summary of information about the best way to do things. A mantra for such a guide might be: 'Don't scare off the readers'" (p. 371). While this recommendation was valuable, she did not go any further in describing the type of content or form that would make these guides useful to inexperienced writers. In effect, both authors seemed resigned to the idea that, in order to create consistency, “when someone [updated] a page, the [professional] technical writer [would] perform a review and tweak the wording and layout” (Maddox, 2012, p. 371). But in reality, the content sometimes required too much editing and rework to make the effort worth it (Johnson, 2012, p. 38).

To discover how we can better support amateur authors on these wikis, I did a broad content analysis of the authors support guides on the five crowdsourced documentation wiki sites analyzed

above. The results of this analysis is summarized in Table 1.4 below.

Table 1.4

Content Areas of Author Support Guide

Wiki	Categories	Contact information	Graphics	Links	Mentoring	New Author Welcome	New pages	Structure of wiki	Templates	Titles / Headings	Wanted pages	Workflow	Writing style
Blender	X	X	X	X		X		X	X	X		X	X
Fedora		X	X ^a		X	X					X	X ^a	X ^a
OpenOffice	X	X	X	X		X	X	X	X	X	X		X
TinyOS	X ^b	X	X ^b	X ^b		X	X ^b			X ^b			
Ubuntu	X	X	X	X		X	X		X	X			X
^a On DocBook site													
^b On Media Wiki Guide													

There were two outliers among these sites. First, the TinyOS wiki did not have a dedicated author support guide. Instead, a brief section at the bottom of its home page covered three topics: 1) contributors needed to create accounts first, 2) those unfamiliar with wikis needed to read the “Wiki User’s Guide,” and 3) the content was licensed under Creative Commons Share Alike (“TinyOS Home Page,” n.d.). The wiki user’s guide mentioned here was an external guide developed for the MediaWiki platform (it was the same guide used for supporting Wikipedia writers). Likely, this strategy worked for the TinyOS wiki because it had a much smaller user and developer base than the other software discussed here. Still, this lack of customized content limited the pertinence of the TinyOS author support to this discussion. For this reason, I will focus primarily on the other four author support guides below.

The Fedora author support guide was also a nonstandard case because the organization published most of their documentation on a crowdsourced DocBook site (i.e., not a wiki). The author support guide for Fedora was spread across both the wiki and DocBook sites, but it was not

always clear which of the two sites the author support was designed for. I took an inclusive approach by including the content of author support on both the wiki and DocBook sites in the analysis.

Four topics appeared in all of four customized guides: contact information, graphics, new author welcome, and writing style. Four other topics appeared in three of the four guides: categories, links, templates, and titles/headings. If we were to define author support guides as a genre, it would include these two sets of topics as obligatory and conventional, respectively.

In the sections that follow, I will discuss the content areas shown in Table 1.4 above, including the optional (or less common) areas. The latter areas were often particularly interesting in how they related to the nonstandard audiences for these guides. In these sections, I will also occasionally offer brief contextualizations and/or critiques of the design of the content areas as necessary.

OBLIGATORY CONTENT AREAS

The obligatory content areas covered typical topics for a style guide (writing style and graphics use), but they also included two sections that met the exigency of a crowdsourced text (community contact and new author welcome).

Contact information. New authors needed to get in touch with the documentation community, so all of the author support guides included some sort of contact information for the group. In some cases, this took the form of a specific contact person. For example, the Blender wiki directed new users to contact the Admin Coordinator. However, several author support guides only provided a link to the mailing list for the documentation group.

Graphics. Nearly all professional texts on creating documentation have included information on using graphics effectively. For example, Weiss (1991) discussed images as a way

to help make documents accessible for readers (p. 155). Low (1994) included a chapter on how to obtain and use diagrams in documentation. And Horton (1994) told professionals to “include graphics in all documents” (p. 289). Since this was an established concern in the creation of effective documentation, it was not surprising that all of the customized guides included a discussion of graphics. However, most guides did not provide useful guidance on when to use graphics: three guides (Apache OpenOffice, Blender, and Ubuntu) only explained the process of adding images (either by giving the wiki markup code or other instructions). Only the Fedora guide provided guidelines on when to add an image. Notably, it recommended not adding images because they caused both confusion (due to differences between the documentation author’s and user’s systems) and also problems during translation (“7.3. Pictures and screen captures,” n.d.).

New author welcome. These documentation wikis relied on attracting new contributors— as existing contributors phased out of working on the wiki, new ones were needed to replace them. All five of the sites asked readers to contribute to the documentation.

Each homepage featured a short request for contributions. This request normally included two elements: 1) a question such as “Want to help out?” and 2) a link to the author support guide. In addition to this initial request, all of the sites also contained an extended statement welcoming users to contribute. This typically occurred on the first page of the author support guide (with the one exception being TinyOS, which placed this statement on the homepage of the wiki).

The main variation in this content area was the tone of the welcome. Some of the sites used a friendly, casual tone typical to correspondence. For example, the Blender guide opened with “It’s very easy to make simple changes to the contents of the Blender Manual wiki, and there is lots of help available (see above list). It only takes a few clicks” (“Writer Guide,” n.d.). This section concluded with “Happy writing!” (“Writer Guide,” n.d.). Other sites used a neutral tone

(which was more similar to the documentation found on the site). For example, the Ubuntu guide opened with “This website is free for all to edit, and contributing is easy. You need an account for the website (see /Registration for details), and it is recommended that you read through this guide before contributing” (“Writer Guide,” n.d.). Overall, there did not seem to be a strong inclination towards either tone: both the Apache and Fedora sites used friendlier tones, while the TinyOS welcome was neutral.

Writing Style. All of the guides discussed writing style. Advice ranged from organization to word choice to formatting. Typically, these style sections were long and used bulleted lists of instructions in the imperative tense. For example, OpenOffice’s guide instructed authors to “Use the PNG format for screenshots” (“Help Style Guide,” n.d.), and Blender’s guide told them to “Be clear and concise” (“Style Guide,” n.d.). These two sites were also the only ones to include examples of effective style.

The OpenOffice guide gave short examples of good style alongside its more general guidelines. For example, it paired the guideline “Use simple declarative and imperative sentence structures;” with the example “‘To change the password, replace the current password’ is better than ‘If you want to change the password, you can replace the current password’” (“Help Style Guide,” n.d.). Unfortunately, this method did not always seem to address itself specifically to the needs of amateur authors. For example, new authors should have understand that the first version of the sentence in the above example was preferred, but they still may not have understood the meaning of the terms “declarative” or “imperative,” or even how to construct sentences in that particular style.

The Blender Wiki took a different approach. Rather than embedding short examples, it first discussed good style (using imperative guidelines) and then linked to pages with “Good,”

“Bad” (ineffective), and “Ugly” (technically acceptable, but not ideal) styles (“Style Guide,” n.d.). (Or at least, it claimed to do so: in fact, there were no links in the “Ugly” section, and only one example in the “Bad” section.) These were not links to sample pages created exclusively for the style guide. Instead, they led to active documentation on the wiki. While this technique had the advantage of offering more robust models, it also may have impaired the usability of the examples in other ways. First, there was no indication on these live pages as to why their style was Good, Bad, or Ugly. Second, since the pages were living documentation, there was also the potential for changes to affect the pages’ style quality. In fact, this was likely since three of the four pages used as examples had been updated more recently than the Good, Bad, Ugly listing.

Overall, even the best writing style content areas left significant room for improvement. In short, these sections of the guides were apparently trying to mirror professional and/or organizational style guides, but they often were not tailored to the needs of inexpert writers. For instance, only two sites offered examples, but these examples assumed a certain level of existing knowledge (or a willingness to turn to outside resources). One potential solution would have been to offer annotated examples of complete (and static or edit-protected) pages so users could see the effects of the style guidelines on the end product.

CONVENTIONAL CONTENT AREAS

The conventional content areas mainly dealt with topics that were typical for style guides for online documentation. However, there was also some focus on the particular exigencies of developing a usable wiki product.

Categories. Categories, a form of navigable metadata used to group similar pages, were one of the primary tools for organizing content on these wikis. In some cases, they were even used to automatically populate pages that listed all members of a category. The documentation wikis

used these categories both to structure the content of the documentation and also to label pages that need further development. Typically the author support on this topic explained how to add existing categories to pages. The OpenOffice guide also described policies for creating new categories: it warned users “Do not create duplicate Categories for the same topic. Each Category must be unique. Otherwise it will be difficult to use the Categories for searching the wiki” (“Wiki Editing Policy,” n.d.).

Links. In general, the author support guides focused on the proper formatting and wiki markup language for links. However, a few also gave style advice. For example, the OpenOffice guide had the most substantial conversation of linking, with advice like “Avoid overlinking,” and “Weave link text into sentence structure” (“Help Style Guide,” n.d.).

Templates. The word “template” had two distinct meanings in the author support guides. In some cases, templates were pieces of wiki markup code placed into existing pages to denote qualities such as a page being edited or a page that needed work (see, “Wiki Editing Policy,” n.d.). In other cases, templates were incomplete documentation pages used to structure new pages (see, for example, “Help On Templates,” n.d.). Regardless of the way that templates were implemented, these building block structures made new authors’ initial contributions less overwhelming and encouraged consistency across the wiki.

Titles and headings. The guides focused primarily on how to format titles and headings in wiki markup language. Sometimes, they also often offered a few guidelines for effective titles or headings. For example, the Ubuntu guide told users to “Avoid using the word "Howto" in the page title. Everything here is a howto!” (“Page Creation,” n.d.).

OPTIONAL CONTENT AREAS

On one hand, the optional content areas might have been less essential to building an author

support guide. On the other hand, many of the more innovative ideas for meeting the needs of new authors were only used on a few sites. Hence, these innovative ideas became “optional” by genre standards. This group of content areas covers a range of topics, the most interesting of which explored the experience of working on a documentation wiki.

Mentoring. Out of all the content areas, this one stuck out, and while only the Fedora guide included this kind of advice, its prominence there demanded attention. Several main pages in the Fedora author support guide directed readers to a page titled “How to be a successful contributor.” The sole purpose of this page was to offer mentoring advice on how to be a good member of the crowdsource community. For example, this section of the guide instructed potential contributors to “Get permission from work and family” (including details on how to do so without upsetting anyone involved) (“How to be a successful contributor,” n.d.). Other sections on this page included: “Time commitment,” “Joining,” “Observation” (of the project in the process of joining), “Pick what you want to work on,” “First contact,” “Find a mentor and sponsor,” “Look for work,” and “Quitting” (“How to be a successful contributor,” n.d.). This type of personal advice was largely absent from the other author support guides. Certainly the other communities might have offered this advice in personal communications after the potential contributors established contact, but there was no evidence of this in the guides. Fedora’s inclusion of this information acknowledged the importance of supporting the lived experience of the authors as well as the end product of their work.

New pages. The proliferation of pages with similar topics had the potential to be a serious problem for documentation wikis. Rather than making usable and strong wiki pages, this practice resulted in dispersed contributions that weakened all of the pages. Both of the guides that explicitly mentioned this topic advised potential contributors to search through existing pages before adding

a new one (“Page Creation,” n.d., “Wiki Editing Policy,” n.d.). Another possible solution to the issue was implemented by the Blender wiki. While they did not mention this topic, they did discuss set wiki structure: “The pagenames and structure for the current version of the Blender User Manual have been the subject of much debate, deliberation and compromise between the User Manual Admin Board and other Blender Community coordinators” (“Writer Guide,” n.d.).

Structure of the wiki. Only two of the author support guides discussed the overall structure of the wiki. The Blender guide focused solely on explaining the site’s fixed structure (that contributors could not change). And the OpenOffice guide helped new contributors understand the purpose of the wiki’s structure: “All documentation pages are subpages of the main Documentation page. A further substructure presents sections as subpages containing further subpages. Think of it as a hierarchical representation, just like a file system” (“Wiki Editing Policy,” n.d.). This guide also included a depiction of the overall site tree with an explanation of purposes of the main sections. In this way, OpenOffice’s guide ensured that new contributors did not make organizational errors from a lack of knowledge. On the other hand, the lack of this content was a fairly major oversight in the other guides. Even if authors did not directly use this information, it could still help them to see their efforts as part of a larger project.

Wanted pages. In all likelihood, there were occasions when documentation wiki users were willing to contribute but did not know what to work on. The OpenOffice author support guide did the best job of meeting this need. It used categories to tag pages that needed help and also those that were being edited. Then it used these categories to create lists of pages for contributors. The Needs Rework category had 51 pages listed (“Category:Documentation/NeedsRework,” n.d.), while the Being Edited category had 53 pages listed (“Category:Documentation/BeingEdited,” n.d.). The guide also included a less effective

Documentation Wishlist that was rarely updated and included only 3 items (“Documentation Wishlist,” n.d.). This difference in quantity demonstrated the utility of using an automated system like the Categories since it required significantly less upkeep by contributors. However, the automated system was only able to focus on improving existing pages, which limited its impact on the overall design of the site. On the other hand, any list of potential pages would need to be attended to more regularly than the current OpenOffice wishlist.

One possibility, in a sufficiently robust community, would be to use a system similar to that of wikiHow (a site for crowdsourced instructions on a wide range of tasks). One of the more intriguing aspects of this site was the request form for a new article. This form was used to populate a long list of potential articles (“List Requested Topics,” n.d.). Of course, there were drawbacks to this system, including numerous requests for pages that already existed. These extraneous requests resulted both in new, unnecessary pages and also in an excessively long list of requested pages that was regularly edited down. Still a similar system could have helped users of documentation to provide recommendations directly to the wiki contributors.

Notably, the Fedora DocBook site included a similar concept. It used Bugzilla to log bugs in the documentation as well as in the software itself. Of course, this bug reporting system did not extend to the wiki, but it still represented a form of feedback on documentation that could have been valuable to wiki contributors.

Workflow. Creating a clear workflow (a repeatable process or pattern of activity that organizes the work of an organization) could have also helped these documentation wikis to address the inconsistencies on their sites. While workflow is often managed behind the scenes (by a project manager), in open communities like those of crowdsourced documentation wikis, this could turn into an expectation that certain segments of the community shoulder a disproportionate

load.

Only two author support guides directly discussed the topic:

- The Fedora wiki had a dedicated section in its DocBook that recommended a process of writing, then formatting, then publishing. However, only the last topic was discussed in any detail (“Chapter 2. Documentation Workflow,” n.d.).
- The Blender guide had a substantial discussion the workflow and tools for creating a new page. This process began with an author committing to a task, communicating this commitment to the mailing list, and then placing an “in progress” label next to the task on the “Wiki Tasks” page (“Wiki Tasks guide,” n.d.). Following this process, the writer developed content on a sandbox page while keeping the community up to date on her progress (“Wiki Tasks guide,” n.d.). Finally, she would get back in touch with the mailing list to gain consensus on including the now complete document in the official documentation (“Wiki Tasks guide,” n.d.).

Overall, author support guides did not dedicate much space to helping contributors understand process of creating effective documentation. Even the Blender author support guide did not discuss each step of the workflow in depth, but it did at least establish and explain a process that used the tools of the wiki to help organize the writing work. This information probably would have been a simple addition for most author support guides. However, the distributed nature of the workers and work activities could have made this kind of information ineffective if it was not accompanied by a system of control similar to that of Blender’s wiki.

CRITIQUE OF CURRENT AUTHOR SUPPORT

While the existing author support guides were extensive resources, they still left significant room for improvement. In the analysis of these guides above, I discussed a few possible changes,

such as providing complete annotated examples of good style, offering additional mentoring support, and improving the descriptions of wiki structure and workflow practices. The common denominator to all of these recommendations is that they try to move the guides towards user-centered design.

The idea of creating a user-centered style guide is not new. Scholarship on developing style guides has repeatedly recommended integrating user research and audience analysis. Washington (1991) described a process for developing style guides that began with gathering information from potential users of the guide. Likewise, Gale (1996) said style guides failed when users were not involved in their development. And, on the related topic of GUI style guides, Torres (2001) argued that the development of style guides needed to follow a more robust user-centered methodology, including steps where stakeholders were directly involved (p. 16). Finally, Warren (2001) argued that understanding the background of the reader was essential to designing appropriate metadiscourse in style guides (p. 165). Although this scholarship has been a useful corrective against seeing style guides as a top-down enforcement of guidelines, I want to open another possibility: documentation design theory could also help us respond to the specific needs of the amateur, distributed authors contributing to these wikis.

It was clear, even in their introductions, that the author support guides conceptualized their users in problematic ways. For example, all of the new author welcomes included a statement asking potential authors to consult the support guide before making any changes to the wiki:

- Apache OpenOffice: “1. Set up a user ID on the OpenOffice.org Wiki. / 2. Read the Wiki Editing Policy. / ... 4. Edit the page and make the changes and edits as needed, and save” (“Contributor’s 101,” n.d.).
- Blender: “To help you with more complex edits (e.g. updating to describe latest Blender

developments, filling missing pages, etc.), please read the Writer Guide below” (“Writer Guide,” n.d.).

- Fedora: “Things to do first: Read how to be a successful contributor” (“How to be a successful contributor,” n.d.).
- Ubuntu: “...it is recommended that you read through this guide before contributing” (“WikiGuide,” n.d.).

Consulting these guides would have been an important step for potential contributors. However, these statements ignored the basic understanding of how users interact with similar texts.

If we accept the theory that literacies are a technology (Ong, 1986), then the unfamiliar, specialized literacies employed by authors contributing to these wikis could be supported in the same way as any other technology. Based on this understanding, we can reconceptualize author support guides as documentation for the technology of technical writing, targeted towards the audience of non-expert writers, and for the contextual work of producing crowdsourced documentation on a wiki. And for more than thirty years, task-oriented design has shown that users will not read documentation linearly and they will not read extensively before taking action (P. Sullivan & Flower, 1986; van der Meij & Carroll, 1998). In this context, it did not make sense to ask new authors to consult an extensive guide before contributing.

Instead, we might reconsider how to design these author support guides by looking at documentation design theories that focus on productive ways to get novice technology users up and running. Perhaps the most famous theory in this vein is minimalism, which has been primarily connected to the work of John Carroll in the late 1980s. At its height, minimalism became a very popular design theory; however, “the methodology [was] often oversimplified and misunderstood (van der Meij & Carroll, 1998, p. 56). In order to combat oversimplification, van der Meij and

Carroll outlined four fundamental principles of minimalism: “choose an action-oriented approach, ... anchor the tool in the task domain, ... support error recognition and recovery, [and] ... support reading to do, study and locate” (p. 21). The following sections explore the implications of applying these four principles to the design of author support guides.

CHOOSE AN ACTION-ORIENTED APPROACH

Minimalism specifically did away with long preambles/introductions. Based on this principle, van der Meij and Carroll (1998) said that documentation should “provide an immediate opportunity to act” (p. 22). In their discussion, they often realized this principle in the creation of tutorials focused on real tasks.

As noted above, author support on crowdsourced documentation wikis ignored this principle by asking authors to read full guides before taking any action. By creating a system that required extensive reading before action, these wikis increased the possibilities that authors would not read the guides at all.

Author support guides using this principle could offer tutorials (or basic exercises) for new authors, provided that these tasks meet the desire of actually contributing. Several options exist for integrating productive tutorials onto crowdsourced documentation wikis:

- Wiki training could include a sandbox page “where people can try out the interface and practice writing wikitext” (Gentle, 2012, p. 100).
- Wiki managers could create a category for pages with typos (which might even be intentionally left on the page) in order to “[give] users a chance to correct small areas before attempting to revise an article or add a new article” (Gentle, 2012, p. 100).
- Wiki managers could create a category for wanted pages focused specifically on short, beginner-friendly pages (such as reference documentation). Experienced users could also

be encouraged to visit pages in this category to offer feedback and encouragement.

Offering these options does not mean that tutorials need to be enforced as a gate keeping measure. Instead, by offering (and promoting) these opportunities, wiki managers could create an environment in which new authors feel confident in contributing early and often.

ANCHOR THE TOOL IN THE TASK DOMAIN

This principle was closely related to the traditional documentation theory of task-orientation (which said that documentation should support users' goals rather than their understanding of a technology). It meant that documentation users needed to be assisted with their real tasks and that the documentation designers had to acknowledge the desire to use the technology for pre-existing goals. The most important aspect of this principle, for our purposes, was the call to organize the information according to task structure. This structure has already been commonly used in another genre designed to support amateur technical communicators: writing textbooks. For example, Anderson's (2014) *Technical Communication: A Reader-Centered Approach* included sections on guiding readers through the process of crafting various genres of reports. Only one author support guide, the Apache Wiki Editing Policy Page, offered this kind of advice, but it was inadequate. It did not lead the authors through different phases of drafting documentation, but, instead, focused solely on creating a title for the page, saving the page, and offering writing style recommendations.

An author support guide based on this principle would be organized to follow the authors through the process of writing documentation. It could start with joining the community or with planning a documentation page. It could end with advice on editing and testing. Along the way, it could be organized into discrete sections to make the information easily and quickly locatable.

This principle might also entail increasing integrated workflow support. For example, a

wiki manager could create a bot that automatically locates new pages and places them in a “needs review” category. Similar bots could be created to categorize pages that have not been updated for long time or for pages that were recently heavily edited.

SUPPORT ERROR RECOGNITION AND RECOVERY

Minimalist design focused heavily on helping users recognize and then work through errors. This was based on the theory that users were most likely to turn to documentation when they encountered problems that they could not work through on their own.

The author support guides ignored this principle: they did not offer any trouble shooting sections. Instead, they seemed hopeful that new authors could be instructed on all topics prior to beginning so that mistakes were not made in the first place.

Given constraints of the wiki system, there may be few dynamic opportunities (such as “on-the-spot error information” (van der Meij & Carroll, 1998, p. 41)) to help authors actively recognize and correct their own errors. However, wiki managers could tap into the power of the community to meet this goal:

- Many wikis already currently use talk pages to discuss possible changes to pages (or reasons for reverting to earlier states). One possibility for supporting error recognition would be to note errors in the talk page but leave them for the original author to fix. This system could then be combined with categories/templates that noted pages needing editing. In short, wikis could use these tools to develop a workflow that helped authors recognize and correct errors. This proofing might also be organized in a similar fashion to WikiHow, which encourages users to perform specific tasks (such as spell checking, rating articles, formatting articles, and patrolling recent changes).
- Another possibility for improving error recognition would be to create an integrated

feedback tool. Wikipedia used just such a tool from 2011 until 2013 (“Article Feedback Tool,” n.d.). This tool asked readers of articles to: 1) rate the article and 2) offer an explanation for their ability to judge the quality of the article. While the tool has been discontinued due to concerns over its usefulness and the management workload (“Article feedback/Version 5/Report,” n.d.), documentation wiki managers may find that similar, low-commitment feedback tools could support their authoring community.

- Finally, author support guides could also offer a checklist of fixes for common errors for new authors to review as they finalize their pages and post them to the site. This type of resource could help authors to self-recognize errors before they go live.

SUPPORT READING TO DO, STUDY AND LOCATE

Van der Meij and Carroll (1998) used this final principle to offer two recommendations: be brief (p. 43) and provide closure (p. 44). The former was not a major issue for these guides since many of the pages were reasonably direct and short. Even if the context were expanded beyond individual pages, this principle only asked that each section be kept as direct and brief as possible. The second element of the principle was less consistent on these guides: many pages did not provide a summary at the end. However, this lack was rarely conspicuous.

An author support guide following this principle would not break significantly from current practice. The main goal would be to continue to offer author support in the form of modular and searchable documentation. The primary change would be ensuring that each section was self-contained and that it did not necessarily require knowledge of the rest of the guide.

CONCLUSION

While I used the theory of minimalist documentation design as my touchstone here, I do not want to suggest that it is the only viable method for improving on current author support

systems. Instead, my main goal has been to illustrate the kind of changes that become possible once we start thinking carefully about how these guides are being used and who is using them. The inexperienced authors contributing to these crowdsourced documentation projects need more incremental assistance, and they certainly will not read through a whole guide before taking any actions.

This redesign of author support guides can have a real impact on the end product. Hopefully, it can help to address the inconsistency that currently plagues documentation wikis. Yet, of course, I do not propose this as a cure-all—the causes of the current discoordination in the genre system are complex and multiple. However, there is still value in reflecting on current practice. And incremental change can still have a positive impact. Applying user-centered document design theories to the author support guides is only one in a series of future changes to social documentation, but it is still an important place to start.

As a whole, this chapter's focus on author support guides in crowdsourced documentation wikis is meant as one part of a larger exploration of the changing role of technical communication practice and scholarship. In this example, we can see that with the ubiquity of online help systems, professional technical communicators are increasingly being called on to design environments conducive to effective technical communication instead of designing the content of the communication itself. And as amateur technical communicators become more inclined to contribute to complex projects, the field must shift further towards discovering ways to meet the specific needs of these authors.

CHAPTER 2 PLAYER VIDEO GAME REVIEWS

“Chased a guy with a rock, with a rock, around a rock. Game Rocks.” (Sparrow_Hawk, 2014)

As of March 2015, the above review was rated as the “most helpful” review of *Rust* (a video game) on Steam (a game distribution website). It was originally posted on January 18, 2014, and in the subsequent year and two months, 15,303 different people used a voting system built into the website to state they found the review helpful.

I’ll return to the import of this particular review near the end of this chapter, but I want to make two quick observations here. First, this review does not follow the standard form of the user review genre: it ostensibly gives very little information about the product, and it doesn’t offer much of a justification for the reviewer’s evaluation. Second, despite the variation from genre norms, this text is highly valued by the community using it. This combination of factors suggests that player reviews of video games could offer a particularly interesting angle for studying the effects of communities on genre variation and emergence. Additionally, by considering the effect of the interface and site, we can also explore the potential for professional communicators to influence the production of user-generated technical communication.

While user reviews may not seem like an important topic for technical communication scholarship, the relevance of this genre for the field has already been established in the literature. Jo Mackiewicz (2008, 2010a, 2010b, 2011, 2014) has regularly explored the genre of user reviews from the perspective of technical communication. In part, she has argued that technical communication needed to pay more attention to online user reviews due to varied ways in which professional technical communicators are now working with user generated content (Mackiewicz, 2014, p. 439).

With that being said, video game reviews in particular have received very little attention

from the field. Mason (2013) briefly mentioned them in her list of genre ecologies in gaming communities (p. 221), but she focused her discussion on other genres (maps and guides). Aside from this one mention, it appears that there is no other discussion of video game reviews in technical communication scholarship.

In this chapter, I seek to create a foundation for the discussion of video game reviews in technical communication scholarship. To accomplish this, I pursue two ends here. First, by analyzing a corpus of 180 reviews from six communities, I establish a generic structure for game reviews that can apply across the whole corpus. Second, I explore differentiation and variation within certain segments of this corpus to establish whether reviews show variations across specific communities (or websites) and/or types of communities (or websites). Ultimately, the chapter seeks to provide a model for understanding the forces that are influencing the localized variation within this genre in order to explore the roles of technical communication in these texts

METHODS

In this chapter, I use a move-strategy genre analysis method inspired by Swales (1990), Bhatia (1993, 2002), and Biber, Connor, and Upton (2007). Several scholars have already developed genre move analyses of reviews. This work started with professional genres of reviews. For example, Motta Roth (1995) analyzed scholarly book reviews in three disciplines in order to establish reviews as an academic genre. However, work on establishing the genre characteristics of online user reviews has only begun recently. Three studies have explicitly developed unique genre analyses of online reviews. Taboada (2011) analyzed movie reviews and described a two-part structure of a descriptive stage followed by an evaluation stage. De Jong and Burgers (2013) also analyzed movie reviews, but they focused on developing a five move structure that allowed them to differentiate between professional and consumer reviews. Finally, Skalicky (2013)

analyzed product reviews on Amazon and compared helpful positive and critical reviews to discover the genre preferences of the community. This study identified nine common moves in the reviews. In the next section, I'll discuss how the move structures described in these articles influenced the development of my own coding system.

DEVELOPMENT OF CODING SYSTEM

Biber et al. (2007) stated that the process of developing a coding system requires 5 steps: 1) "Determine rhetorical purposes of the genre," 2) "Determine rhetorical function of each text segment in its local context," 3) Group steps, 4) "Conduct pilot-coding to test and fine-tune definitions for move purposes," and 5) "Develop coding protocol with clear definitions and examples" (p. 34). In general, I followed this process when developing my coding system, however I made a few changes that deserve a brief explanation.

First, I encountered some difficulty in defining a clear purpose for the genre of video game reviews prior to coding. The most apparent purpose for an online user review is to provide information on and an evaluation of a product or service to potential users. But as Hennig-Turau Gwinner, Walsh, and Gremler (2004) showed, authors of user reviews have a number of reasons for using the genre, including such things as seeking advice, seeking economic incentives, and seeking to vent negative feelings (p. 46). Likewise, I found that the purpose of the video game player review genre in the sample wasn't necessarily uniform. For example, the following was a complete review for *Sid Meier's Railroads* on Steam:

Review 31: "I like Trains."

The above review also includes an embedded YouTube video by asdfmovie called "I LIKE

TRAINS.¹⁰” One possible approach to this text would be to try to read a traditional review purpose into it. For example, we might say the author is actually referring to the game being reviewed and that the statement is therefore actually a claim about the quality of the game. In this reading, we could maintain the general purpose of the genre and say that the review is seeking to convey information about the product (specifically, its high quality) to the reader. However, how would we then account for the “helpfulness” of this review? This was ranked as the most helpful review of *Railroads* by Steam users, but Steam contained player reviews that provided far more useful information than this. Another possible approach to this review would be to say that it falls outside of the genre. As Swales (1990) also said, the simple fact that a communication is labeled by an institution as belonging to some particular genre does not necessarily make it so (p. 55). In other words, we might say that though the above text was listed among the reviews for a game, it was not actually a video game review.

However, in reading through the reviews in the sample, I didn’t find either of the approaches described above to be terribly productive. While Review 31 was an extreme case, several other reviews did not fit within generally accepted purpose for reviews. Furthermore, moves within reviews often seemed to primarily serve a different end. For example, reviewers sometimes used reviews to offer feedback to the game designer. This move was common enough that it seemed to support an accepted genre purpose, which was, simultaneously, outside of the traditional conception of user reviews in general. For this reason, I accepted that the genre might have a range of purposes and read the sample inductively rather than trying to establish a purpose beforehand. Overall, this deviation in method seems in line with Swales’ (1990) statements that

¹⁰ The video was posted to YouTube years before the review and was (almost certainly) created by someone other than the author of the review (I can’t necessarily prove this because I don’t have access to the real name of the review author).

“purposes of some genres may be hard to get at” (p. 46) and that “it is not uncommon to find genres that have *sets* of communicative purposes” (p. 47).

My second detour from the method described by Biber et al. was using existing coding systems for similar genres to build my initial set of codes. As mentioned above, three studies had already created coding systems for user reviews. The moves identified by these studies are depicted in Table 2.1 below.

Table 2.1

Existing User Review Coding Systems

Taboada (2011) (movie reviews) ¹¹	De Jong and Burgers (2013) (movie reviews)	Skalicky (2013) (consumer product reviews) ¹²
1) Descriptive Stage a) Subject Matter b) Plot c) Characters d) Background 2) Evaluation Stage	1) Giving practical information about the movie 2) Describing the movie 3) Placing the movie in context 4) Giving criticism 5) Recommending the movie to the reader	1) Evaluation move 2) User information move 3) Title move 4) External information move 5) Overall statement move 6) Personal experience move 7) Comparison move 8) Background move 9) Refer to other reviews move

I combined these existing systems to create an initial (unwieldy) coding system. I used this system while reading through an external sample of 25 video game reviews and added new codes when I encountered moves that weren't covered by the initial system. By this point, my system included 50 codes for the various moves and steps that I'd encountered. I then began a process of consolidating and refining the coding system while also engaging in recursive readings of the reviews within the sample. Ultimately, this resulted in a coding system with 5 moves and 14

¹¹ Taboada uses Eggins' (2004) concept of stages rather than Swales' (1990) concept of moves. While there is some difference between these concepts, the structure described by Taboada is similar enough to those of the articles to be included in this table.

¹² Unlike the other two studies in this table, Skalicky (2013) chose to order his moves according to the overall frequency of the move in the corpus because he found that the order of the moves was highly variable in the texts (p. 86).

strategies as described below.

During this process, I aimed to create a coding system that could describe a consistent generic structure to the video game review. According to Biber et al. (2007), “move analysis proposes that genres are composed of definable, and to a great extent, predictable functional components—that is ‘moves’ of certain types” (p. 32). In this context, it seemed important that my final coding system could account for this generic consistency. At the same time, I wanted a detailed enough system that variance within the genre wasn’t obscured. For this reason, my final coding system had relatively few moves and a more detailed system of strategies beneath these moves. My final coding system included five moves described in Table 2.2 below.

Table 2.2

Moves in Video Game Player Review Genre

	Move Name	Move/Strategy Description	Examples of Move/Strategy
1	Background Information Move	Provides context to understand discussion of game in review.	Review 133: "Don't Starve is a survival game"
2	Specific Aspects Move	Describes or criticizes specific aspects or features of the game being reviewed.	Review 40: "Rollercoaster Tycoon 3 Platinum has decent audio (the thrilled screams of children on the rides, the chaching of purchased food and gifts) and pleasant 3D visuals that become a bit ew when zoomed in too far"
3	Overall Quality Move	Summarizes evaluation of game.	Review 21: "This little game is a pure masterpiece"
4	Game Narrative Move	Narrates a story from personal or imagined game experience.	Review 125: "I ran into the first friendly player I had met in Rust. I had initially spotted him pretty far away, but I paid no real notice to him as he had a rock equipped. Eventually he got close enough for us to clearly see each other and, upon the realization that neither of us were hostile, we both pulled out torches as a mutual sign of peace"
5	Advice Move	Offers advice on game to imagined reader (typically either a game player or game designer).	Review 95: "Don't let the "ending" fool you. There's more. Find the Journal. Open the box. It's not too late. You know what to do"

Finally, I applied this coding system across the full corpus of reviews in the sample. This is where I made my final divergence from the method described by Biber et al. Traditionally, a move can be coded multiple times in the same text. However, after an initial coding of the sample, I found that some reviews repeated moves several times and that this repetition caused problems for establishing accurate comparisons between the different websites in the sample. For this reason, I report only whether or not a move is present in a review (i.e., not how many times it is used).

This system of analysis allowed me to more accurately depict and compare variations in the genre structure between different sites.

SAMPLE

The sample for this analysis consisted of a total of 180 video game reviews from six sites (30 per site). The sites belonged to two broad revenue models: digital distribution sites that sold games (Desura, GOG, and Steam) and game review/news sites that relied primarily on advertising for revenue (GameSpot, Giant Bomb, and Metacritic).

The reviews in the sample covered 36 total games from 6 genres. The genres were selected to avoid privileging any specific segment of the gaming community. The six genres were Roguelike, Hard Platformer, Survival, Economy Management, Survival, and Puzzle games. Broadly speaking the first three genres were more “gamer-centric” while the latter three appealed to broader audiences. After the genres were selected, a different game representing the genre was selected for each site. Ideally, the study could have compared reviews of the same across all six sites. However, it was impossible to locate even a single game that had a requisite minimum number of reviews (five) on all six sites.¹³ For this reason, a different game was chosen for each site in order to not create artificial similarities between some sites and not others. This resulted in the aforementioned 36 games (6 genres times 6 sites). Then the top five “most helpful” reviews (as rated by the sites’ communities) were selected for the sample. The full breakdown of the reviews sampled (along with the authors, games, genres, and sites) can be found in the Appendix.

A brief overview of sites can be found in Table 2.3 below. More detailed descriptions of

¹³ This inconsistency makes sense in context. Two of the advertising revenue sites (Game Spot and Giant Bomb) had relatively low numbers of player reviews overall, and they had even fewer reviews of games that were not AAA releases. At the same time, two of the digital distribution sites (GOG and Desura) did not specialize in or carry many AAA release games. Finally, a significant number of games on Steam are purposefully kept as exclusive to that platform.

the sites follow the table.

Table 2.3

Overview of Websites in Sample

Website	Revenue Model	Avg. # of Reviews*	Avg. # of helpfulness ratings	Avg. Word count of review*
Desura	Digital Game Distribution	390	22.7	83
GameSpot	Advertising	76	2.9	476
Giant Bomb	Advertising	23	2.0	666
GOG	Digital Game Distribution	52	90.1	204
Metacritic	Advertising	34	4.9	210
Steam	Digital Game Distribution	12,253	3441.7	153

*Numbers based on sample only and may not be representative of the sites as a whole.

DESURA

Desura was a “community driven digital distribution service for gamers” (“About Us,” n.d.). Desura described itself as “developer driven” (meaning that they tried to help developers present their products well) and “community run” (meaning that users could contribute a range of media to the product pages). In many ways, Desura was similar to Steam (discussed below), but it featured far fewer AAA game releases. Instead, most of the games on Desura were developed by small or indie companies. Additionally, many games on Desura were available DRM Free. Overall, Desura had the second most reviews per game out of any of the sites in the sample. However, it also had the shortest reviews on average.

GAMESPOT

GameSpot was one of the best known video game news and review sites, and it was fairly standard for that kind of site. Its content included reviews, videos, news, and discussion forums. While it focused on AAA titles, reviews of some indie releases were also present. Probably due in part to the lack of a ceiling for review length, GameSpot had, on average, the second longest reviews of any of the sites. It also had the most reviews of a game of any of the non-retail sites. However, like the other advertising revenue sites, GameSpot showed relatively low levels of user

interaction with player reviews as its average number of helpfulness ratings was the second lowest overall.

GIANT BOMB

In 2008, Giant Bomb was founded by a former editorial director at GameSpot. The initial goal of creating the new site was to make “a fun video game website” with “opinionated reviews” (“X-Play Gaming Update,” 2008). From the start, the site also had the intention of letting users add their voices. Then in 2012, Giant Bomb was purchased by CBS Interactive, the same corporation that owned GameSpot. While Giant Bomb has maintained much of its original intent over the subsequent years, the connection to CBS Interactive has also had a significant impact (for example, it now uses the exact same player review system as GameSpot). Giant Bomb not only featured the user longest reviews on average, but it also had the single longest review in the sample (and 6 of the 7 longest reviews). Despite this fact (or perhaps because of it), Giant Bomb reviews also featured the lowest number of usefulness ratings in the sample. Over half of the Giant Bomb reviews in the sample (16 of 30) had 1 or fewer helpfulness ratings.

GOG (GOOD OLD GAMES)

GOG, as its name suggests, focused primarily on selling re-releases of old games. While GOG did distribute some new releases, all of the games in its catalog were DRM free. Like Desura, GOG lagged behind Steam in its overall user base. It had the fewest and longest reviews of any of the digital distribution platforms. However, it had the second most helpfulness ratings overall in the sample.

METACRITIC

Metacritic was a review aggregator: it collected expert reviews of media from various publications and websites and created a score that summarized the opinions of all of the reviews.

Unlike the other sites discussed here, it covered a range of media including movies, games, television shows, and music. Of the advertising revenue sites, Metacritic had, by far, the shortest reviews and also the highest average number of helpfulness ratings.

STEAM

Steam was the largest digital distribution platform for games. It was originally released by Valve, a video game publisher, as a way to update its own games, but had subsequently grown into one of the predominant ways that PC gamers purchased and played games. By February 2015, Steam had “over 125 million active accounts worldwide,” and it offered “4500 games, with 400 million pieces of user-generated content contributed by members of the Steam Community” (Smith, 2015). Given the size of its community, it should come as little surprise that Steam had the most reviews and helpfulness ratings of any of the sites in this study. To ensure that this was not a biased number (i.e., that the most popular games had inadvertently been selected for Steam), I checked the number of reviews on Steam for every game in this sample. On average, Steam had over 6000 more reviews for each game than the other sites. And the most reviewed games on Steam had a truly staggering number of reviews. For example, DayZ (a game that only had 8 user reviews on GameSpot) had close to 70,000 reviews on Steam.

BASIC DIFFERENCES BETWEEN REVENUE MODELS

We can start to notice a few basic trends in Table 2.3 above. Overall, user reviews were posted in far higher numbers to digital distribution platforms than they are to advertising revenue. At the same time, these reviews were also interacted with more by the community. On the other hand, users posted shorter reviews to digital distribution sites. All three of these trends were not coincidental, but instead related to how users interacted with each type of site. Reviewers on digital distribution sites were likely already interacting with those sites in the process of purchasing and

playing the games. Reviewers on advertising revenue sites, on the other hand, would have had to search out those sites after playing the game, suggesting that their reviews were likely planned out in greater detail. I'll return to this discussion later after exploring the variation in the genre in more depth.

STUDY LIMITATIONS

There were a few limitations to this study. First, the sample was not ideal because it was impossible to compare set games across all six sites. While the use of common game genres likely created some consistency, the specifics of individual games had an impact on how reviews were written. This will be discussed in more detail later. Future studies could create stronger controls, most likely by studying reviews of the same game across a smaller sample of sites.

Additionally, since this study was positioned as a part of my dissertation, it is a single coder system without any proof of cross-coder consistency. Ideally, this study could be replicated in the future with a larger corpus and with multiple coders.

RESULTS

The five moves, as described in Table 2.2 above, covered the means of achieving common rhetorical goals in all of the reviews in the sample. However, they were not used with equal frequency, as shown in Table 2.4 below.

Table 2.4

Overall Frequency of Moves

	Move Name	Frequency of Move in Sample
1	Background Move	127 (70.6%)
2	Specific Aspects Move	129 (71.7%)
3	Overall Quality Move	176 (97.8%)
4	Narrate Game Experience Move	10 (5.6%)
5	Give Advice Move	25 (13.9%)

Given the disparity in the frequencies, Moves 1-2 were labeled as conventional (>60%), Move 3 was labeled as obligatory (\approx 100%),¹⁴ and Moves 4-5 were optional (<60%) as defined by Kanoksilapatham (2005).

Additionally, the order of the moves showed a relatively consistent pattern in the reviews. The most common moves for the opening and closing of the review are shown in Table 2.5 below.

Table 2.5

Frequency of Moves as Opening and Closing of Reviews

	Background Information Move	Specific Aspects Move	Overall Quality Move	Narrate Game Experiences Move	Give Advice Move	Total¹⁵
Opening	90 (50.3%)	22 (12.3%)	61 (34.1%)	6 (3.4%)	0 (0.0%)	179
Closing	15 (8.4%)	25 (14.0%)	126 (70.8%)	1 (0.6%)	11 (6.2%)	178

There was more variance in the review opening than in the closing. The overall quality move was used in a strong majority of all closings. The background information move was only used as an opening move in a half of the reviews. Regardless, it was used at a far higher rate than any other move to open reviews.

The most common generic move structure was by far Background Information Move—Specific Aspects Move—Overall Quality Move. An example of a (short) review that follows this structure is shown in Table 2.6.

¹⁴ According to typical definitions of obligatory in move-strategy analysis, Move 3 would need to be present in 100% of the reviews. However, one might make a claim that it is if we were to expand the focus to include the rating system in addition to the content. Furthermore, the four reviews that do not make an overall statement move in the written content of the review are extreme outliers as will be discussed later in the section on the Overall Quality Move.

¹⁵ Note that there were 180 reviews in the sample. The openings and closings do not quite equal 180 in this table because a few opening and closing reviews did not fit into the move definitions used in this analysis.

Table 2.6

Review 92 Separated by Moves

Move 1 (Background Information):	“It's been months since I first played Year Walk on iOS and I still think about it regularly. The Steam version of this game is as beautifully made as the original, and the changed puzzles feel at home on the PC.”
Move 2 (Specific Aspects):	“The atmosphere is greatly enhanced in this version of the game, providing a creepier mood than the iOS version - so much so that surprises I knew were coming still got jolts out of me. The story behind this game is so intriguing and mysterious that I wish there was more information about it online;”
Move 3 (Overall Quality):	“Year Walk is an amazing introduction into Scandinavian folklore that everyone should play.”

While this complete structure occurred in only 70 of 180 reviews (39%), it still represents the conventional generic structure for this highly variable genre.

The second most common structure was a review that both opened and closed with an overall quality statement. This pattern occurred in 35 of 180 reviews (19%), but this number was largely influenced by single move reviews on Desura and Steam. These reviews, which consisted of only a single move, accounted for 18 of the 35 reviews following this structure.

In order to discover the differences between the sites with different revenue structures and between individual sites, I calculated the frequency of each move and strategy on each site. Then I used chi-squared tests (χ^2) to find moves and steps significant differences between the observed frequencies and an equally distributed frequency for all sites.¹⁶ Tables 2.7 and 2.8 below show the results of this analysis for the moves on revenue types and individual websites respectively. Appendices B and C show the full results for both moves and strategies. I will discuss these results further in the next section.

¹⁶ Chi-squared tests show the probability of a data distribution occurring randomly. Results are generally considered significant if the result of the chi-square test is $>.05$.

Table 2.7

Comparison of Move Frequency by Site Revenue Type

Move Name	Advertising	Digital Distribution	Total	χ^2
Move 1	76	50	126	0.02
Move 2	72	57	129	0.19
Move 3	89	87	176	0.86
Move 4	2	8	10	0.06
Move 5	13	12	25	0.84
Total	243	212	455	0.15

The only significant finding in the frequency of moves by site type was the Background Information Move, which was used by advertising revenue sites far more often than digital distribution sites. There was also a marginal result in the frequency of use of the narrate game experience move. I'll discuss each of these further in the next section.

Table 2.8

Comparison of Move Frequency by Website

Move	Desura	GOG	GameSpot	Giant Bomb	Metacritic	Steam	Total	χ^2
Move 1	14	21	24	26	26	15	126	0.20
Move 2	16	27	27	23	22	14	129	0.22
Move 3	29	30	30	30	29	28	176	1.00
Move 4	0	0	1	0	1	8	10	0.00
Move 5	2	6	7	6	0	4	25	0.12
Total	61	84	89	85	79	69	467	0.19

The only significant variation in move frequency across the individual websites was for the narrate game experience move. However, there were a number of additional significant variations at the level of strategies that I will discuss in the next section.

DISCUSSION

In the sections that follow, I will discuss the moves (and their strategies), while focusing on the variation of genre construction on different sites. Then I will discuss how individual games might have influenced some of this variation. Finally, I will also discuss the way that the gaming community has reacted to some of the variation.

BACKGROUND INFORMATION MOVE

As mentioned above, the Background Information Move was the most common opening move in the corpus. It was used to frame the evaluation of the game and the statements about specific features of the game. Overall, this move was present in 126 of 180 reviews. 90 of those 126 instances occurred when the Background Information move was the opening move of the review. However, there was some inconsistency in the use of the move overall. Table 2.7 above shows that there was a significant difference in the frequency of the move between the advertising revenue driven sites and the digital distribution sites. In what will become a recurring theme through this discussion section, much of that difference was driven by the short reviews present on the latter group of sites. There were a total of 40 reviews in the corpus that were less than 50 words long. Overall, digital distribution sites had 30 of these 40 reviews (Desura-15; GOG-1; Steam-14). And only 15 of these 40 (37.5%) used the background information move, which was a far lower rate than the corpus as a whole (70.6%). However, the numbers became even more interesting at deeper levels of analysis. 8 of 10 (80%) advertising revenue site game reviews under 50 words still used the background information move—this was a frequency similar to that of the whole sample from advertising revenue sites (87.8%). Short reviews on digital distribution sites had a far lower frequency of use of this move (23.3%). If we flip the focus though, and look at reviews longer than 50 words on digital distribution sites, 53 of 60 reviews used the background information move—a rate (88.3%) that was much more in line with the sample from advertising revenue sites. The short way to say this is that the difference between site types here was a variant form of the review genre that appeared on digital distribution sites but not on advertising revenue sites. This type of review didn't seek to offer context for its evaluation; instead, it simply stated an evaluation quickly and directly.

I coded for three common strategies in the background information move. These strategies are defined in Table 2.9 below.

Table 2.9

Description of Background Information Strategies

	Strategy Name	Strategy Description	Example of Strategy
1a	Place game in context	Relates game to other existing games. Most often a reference to a genre of game or to specific similar or dissimilar games.	Review 139: "This is Minecraft (with much better graphics) combined with 'The Walking Dead'"
1b	Establish reviewer's gaming experience	Provides information on reviewer's experience/expertise in gaming. Most often a reference to play time on game being reviewed.	Review 63: "Over 240 hours, 1500+ runs, I've completed this game a measely 12 times"
1c	Place review in context	Provides context specifically about reviews or the reviewer. A range of tactics exist include implicitly discussing genre of reviews and discussing other reviews of game.	Review 90: "Granted, reviews are supposed to be one's opinion which lets people determine if the game's worth their cash but when a game is praised this much, kind of makes the one review stick out"

Strategies 1a and 1c followed a similar pattern to the overall frequency of the move across site types: they were used far more frequently by advertising revenue sites than on digital distribution sites (see Appendix 2).

However, as shown in Appendix 3, an interesting pattern emerged in how strategy 1c was applied on individual sites. Reviewers on Giant Bomb put their own review in context at a far higher rate (12 of 30 reviews, 40%) than reviewers on any other site (highest on other sites was 6 reviews, 20%). Reviewers on Giant Bomb used a few techniques to accomplish this strategy. In some cases, they contextualized their review by discussing the genre of reviews in general. For example:

Review 90: "Granted, reviews are supposed to be one's opinion which lets people determine if the game's worth their cash but when a game is praised this much, kind of makes the one review stick out."

In other cases, they contextualized their review by putting it into conversation with existing reviews of the game. For example:

Review 117: “You don’t need me to tell you that Telltale’s The Walking Dead is a big deal. Since its release in 2012, the game has received a slew of accolades from gaming press and fans alike”

The fact that Giant Bomb reviewers felt the need to contextualize their reviews in this way showed that the genre expectations on that website may have been slightly more developed than on other sites. The reviewers were likely expecting readers who had experience in reading reviews and had already read reviews on other sites. By contextualizing the review itself, the Giant Bomb authors were acknowledging this existing knowledge base and were using it to develop nuanced approach to the evaluation of the game.

Returning back to the larger view, the one strategy that didn’t follow the overall pattern of the move was 1b (establish reviewer’s gaming experience). This strategy was used at effectively the same frequency in digital distribution (24) and advertising revenue sites (23). This break from the pattern suggested that this type of contextualization is particularly important on digital distribution sites. While determining the exact rationale behind this usage rate was impossible, the difference was most likely attributable to the interface of the websites. Of all the sites, reviews on Steam were by far the most likely to select strategy 1b to accomplish move 1. Overall, Steam had 15 reviews that used the background information move, and 10 of those reviews used the strategy of establishing the reviewer's gaming experience. While this use frequency wasn’t out of line with other sites (both GOG and Giant Bomb also had 10 reviews that used the strategy), the difference in proportional use still suggested that Steam reviewers found this strategy particularly important. However, by taking a deeper look, it became clear that many of the uses of this strategy on Steam were references to the number of hours that the reviewer had played the game. For example:

Review 63: “Over 240 hours, 1500+ runs, I’ve completed this game a measely [sic] 12 times.”

At the same time, Steam was the only site that publically listed authors’ total number of hours in the game next to their reviews. This may have encouraged some authors to make self-referential statements about how much they had played the game, particularly if some of this play time occurred off of the Steam platform. If nothing else, this variation showed one way an interface might be used to encourage particular types of statements within the genre.

SPECIFIC ASPECTS MOVE

Overall, the specific aspects move was the second most common move in the corpus. It was used in 129 of the 180 reviews (71.7%). This was logical since most reviewers needed to reference specific features of a product in order to explain their overall evaluation.

The specific aspects move followed a similar pattern to the background information move. It was commonly used in reviews on advertising revenue sites (72 of 90 reviews, 80.0%) and at a lower rate on digital distribution sites (60 of 90 reviews, 63.3%). Just like with the background information move, very short reviews (>50 words) had a low rate of use of this move (13 of 40 reviews, 32.5%). And again, reviews from digital distribution sites within this sample had an even lower rate of using this move (8 of 30 reviews, 26.7%). And again, longer reviews on digital distribution sites used this move at a similar rate (81.7%) to longer reviews on advertising revenue sites (83.8%).

As shown Table 2.5 above, this move was not commonly used as an opening (22 of 179 reviews, 12.3%) or closing (25 of 178 reviews, 14%) for the review. When it was used as an opening, it was most often because a reviewer had a specific complaint about the game. For example:

Review 98: “I am experiencing some serious bugs”

However, there were also cases when the specific aspects move was used as an opening in order to preview specific topics that would be discussed in more depth later. For example:

Review 26: “The Binding of Isaac has at least three hooks: its unique gameplay, its procedural generation which encourages repeated playthroughs, and its very distinctive aesthetic with a sinister sense of humour”

When the specific aspects move was used as a closing, it often followed an overall quality move.

In these cases, the specific aspects move was used to reinforce some particular claim about why the game as a whole was succeeding or failing. For example:

Review 137: “What they actually need is a Graphic and Performance Improvement”

I coded four strategies in the specific aspects move. The four strategies are defined in Table 2.10 below.

Table 2.10

Description of Specific Aspects Strategies

	Strategy Name	Strategy Description	Examples of Strategy
2a	Discuss gameplay	Describes or criticizes gameplay or design elements including graphics, control scheme, difficulty, etc.	Review 173: "As you move through the world colored blocks appear. Players manipulate these colored blocks to aid themselves or move objects to a destination. Special gloves allow you to activate the blocks from a distance or even bestow neutral blocks with a specific block type"
2b	Discuss practical aspects	Describes or criticizes practical aspects of game including price, stability of platform, system requirements, etc.	Review 150: "Shadow of Chernobyl costs \$20 on Steam, oddly enough making it more expensive than its sequels"
2c	Discuss game plot	Describes or criticizes elements of game's plot, including characters, events, general goals of story, etc.	Review 85: "It is the story of a young man, Tim, and his journey to find a beloved princess"
2d	Discuss developer or development practices	Describes or criticizes the game's developer or development practices.	Review 138: "The developers are now more focused on body odor and bathing mechanics, and making sure the female characters look sexy in their g-string underwear when they're "disrobed". Seriously?!? They'd rather offend half the gaming community instead of adding real stuff to do?"

While these four strategies showed several interesting usage patterns, many of these seemed to be influenced by the exigence of how communities responded to specific games: in other words, the use of a strategy was concentrated around reviews of specific games. I'll discuss this pattern in more depth later.

For now, I only want to focus on one strategy pattern: discussion of the game's plot (2c). 22 of 30 reviews on Giant Bomb (73%) mentioned the plot of the game being reviewed. For example:

Review 117: "In the game you play as Lee Everett, a convicted murderer on his way to jail, when the police car he is riding in hits the reanimated corpse of a man and swerves off of the road. Escaping from his fate as a prisoner, Lee gets another opportunity at a

free life, but in a world where the dead are coming back as “Walkers” and civilisation is in tatters”

In general, it could be assumed that reviews certain genres of games (e.g., adventure games) would have been more inclined to include descriptions of the plot. However, every game in the sample from Giant Bomb had at least one review that discussed the plot, even it was to say that there wasn't much plot. For example:

Review 87: “Super Meat Boy does have a narrative, but in true platformer style it is kept to a minimum: Meat Boy’s girlfriend Bandage Girl has been snatched by Dr. Fetus (literally a fetus in a glass robot suit) and you must get her back”

No other site had such a concentration on the plot of a game. Even reviews on GameSpot (which were of similar length and complexity as those of Giant Bomb) only mentioned plot in 13 of 30 reviews (43.3%). This variation on Giant Bomb suggested that game reviews could be influenced by the communities on individual sites. These communities might have established interests in different aspects of games that encouraged reviewers to cover those topics when evaluating the game.

OVERALL QUALITY MOVE

The overall quality move was the most commonly used move (176 of 180 reviews, 97.8%), and it was also the most evenly spread across both revenue types ($s = 1.4$)¹⁷ and individual sites ($s = .82$). Earlier, I labeled the overall quality move as the only obligatory move in a video game reviews because it was ubiquitous in so much of the corpus: in fact, on three of the sites, this move was present in every single review, and these were the only instances of a move being present in every review on any site.

The four reviews that did not use the overall quality move were each an extreme outlier in the sample. They were also all very short (the longest was 46 words long). Table 2.11 below

¹⁷ Standard deviation of the sample

contains the entirety of each of these reviews.

Table 2.11

Reviews without Overall Quality Move

Review ID	Content
31	I like Trains. [This review also had a link to a comedy video about trains.]
44	Game is virtually unplayable after getting to the second stage. Instead of comparing all your games made to your highest scoring game, you should compare it to other simulated companies within the game. Once you have a smash hit game, there's nowhere to go but down.
47	TBH i haven't played it before nor have I watched any videos or have any idea what you actually do in it. I haven't played any of the other Farming Simulator games either.
123	A lot of naked men throwing rocks and trying to survive against zombies, animals and other naked men.

Two of the reviews (31 and 47) were just jokes about the games. The other two were short and maintained a very tight focus on particular features of the games (though review 123 was also intended as a joke).

On one hand, it would have been possible to label these four as “not reviews”—a possibility I already rejected as unproductive earlier. On the other hand, all four of these reviews still made a summary claim about the quality of the game in their required ratings (e.g., 5 stars), so if the analysis had been expanded beyond just the text of the review, this move would have coded as universal. At the same time, we might also recognize that these outliers (particularly the ones that function as jokes) were part of the expanding use of the review genre to discuss games with others who have played the game (a topic I’ll return to in the conclusion).

The overall quality moves was most often used as the final move of the review, though it also accounted for a little over one third of the opening moves. However, the overall quality move was less common as an intermediate move between the opening and closing.

Among short reviews, some only used one move (though they still sometimes used multiple

strategies). For example:

Review 135: "ABSALUTLY [sic] AMAZING, A MUST BUY!"

This type of review often used the overall quality move: 13 of 16 single move reviews used the overall quality move (1 used the background information move and 2 used the specific aspects move). Typically, these reviews were very short: 12 of these 13 overall quality single move reviews were under 25 words long. All of these reviews were posted to either Steam or Desura (which was predictable since these reviews didn't meet the minimum length requirements for GameSpot, Giant Bomb, or Metacritic).¹⁸

I coded for three strategies in the overall quality move. These strategies are defined in Table 2.12 below.

Table 2.12

Description of Overall Quality Strategies

3a	State general quality of game	Offers a general statement about the quality of the game as a whole. Often a statement about the game being "good" or "bad."	Review 146: "S.T.A.L.K.E.R is just one of those games where everything comes together perfectly"
3b	Make purchase recommendation	Instructs the reader to purchase or not purchase the game. Often has only an implicit evaluation of the quality of the game.	Review 163: "If you like puzzles, exploring, programming and/or creating your own solutions, then you owe yourself a copy of this game."
3c	Summarize quality in rating	Uses an explicit rating on a 5 or 10 point scale to state the quality of the game. (In written body of review only.)	Review 156: "Five stars, no doubt."

Neither strategy 3a nor 3b showed significant variation based on individual sites nor revenue type.

Both of these strategies were also very common overall.

The most interesting strategy variation was 3c. This was a fairly uncommon strategy

¹⁸ On the other hand, GOG.com did have some incredibly short reviews (>5 words long), but none of them were rated highly enough to make it into the corpus.

overall, likely because five of the six sites required reviewers to rate the game either on a 10 or 5 point scale when the review was posted. The one exception to this requisite rating system was Steam, which only allowed reviewers to Recommend or Not Recommend a game. Regardless, the usage rate of this strategy on Steam was still consistent with that of other sites. However, a closer look at exactly what is being said on Steam showed another trend in the discourse of these reviews. The four rating statements from Steam are displayed in Table 2.13 below.

Table 2.13

Steam Reviews' Use of Summarize Quality in Rating Strategy

Review ID	Rating Statement
32	BEST GAME EVER. 20/10
65	One of my top 5 games of all time easily 11/10
122	Oh, and I hate this game. 10/10
151	10/10 Best capitalism simulator

In each case, the rating was directly adjacent to a statement of the game's overall quality. And each of these statements was glowing (if facetious) praise for the game. The use of the rating system here is actually one part of a larger shared discourse by this community of gamers, a discourse that does not appear on the other sites in the sample: namely, meme speak. This tendency has led to some criticism of reviews on the site (see, for instance, Valentaten, 2015), which I will discuss later.

NARRATE GAME EXPERIENCE MOVE

The final two (optional) moves were used with far less frequency than the first three moves. The main reason these two were still labeled as moves was because they represented patterned rhetorical goals that could not be classified into the other moves.

The narrate game experience move was the least used move in the corpus. It was also confined almost wholly to Steam (8 of 10 reviews move were on Steam).

At the same time, narrativity in reviews has been of repeated interest in the scholarship on

reviews. Skalicky (2013) included a “personal experience move” in his coding system and found that narratives were used more frequently in negative reviews (p. 87). Based on this information, he argued that negative narratives were more useful to readers than positive narratives (Skalicky, 2013, p. 87).¹⁹ Hamby, Daniloski, and Brinberg (2014) studied the effect of narrativity in reviews on transportation (the feeling of cognitive and emotional engagement with a text), and they found that narratives in reviews increased persuasion (p. 4). Finally, Jurafsky, Chahuneau, Routledge, and Smith (2014) studied narratives in reviews and found that the framing mechanisms used by reviewers were tied to existing narrative patterns: for example, negative reviewed mirrored patterns of trauma narratives (in other words, reviewers framed their negative experiences as essentially traumatic).

The frequency of this move was so low in this study that there wasn’t necessarily a pattern in how narrative was being used. However, there was a significant concentration of the move around reviews of a single game: Rust. Three of these four narratives were intentionally humorous in nature, but they also framed this humor in a way that spoke directly to insiders (i.e., to others who were also familiar with the game). I’ll return to a discussion of these narrative reviews on Steam at the end of this chapter.

OFFER ADVICE MOVE

Finally, the offer advice move was an odd collection. This was the move I was most conflicted about in coding because it merged two distinct audiences (and thus rhetorical ends). However, the presentation of the move was consistent regardless of intended audience.

¹⁹ I have some slight concerns about this conclusion as his sample only included the most helpful positive and critical review for each product. This sample isn’t able to capture whether observed trends were a direct result of helpfulness or whether they actually resulted from trends in how reviewers wrote positive or negative reviews. In other words, only a larger corpus that included both helpful and unhelpful positive and critical reviews could have established the qualities to which review readers were responding.

This move captured moments when reviewers were using their experience with a game to make suggestions to readers.²⁰ There wasn't any significant variation in the frequency of the move across revenue types (see Appendix 2) nor across individual websites (see Appendix 3). The move was slightly concentrated on three of the six sites, but this was still within the possibilities of random variation.

I coded for two strategies in the offer advice move. These strategies captured the different audiences for the advice. The two strategies are defined in Table 2.14 below.

Table 2.14

Description of Overall Quality Strategies

	Strategy Name	Strategy Description	Examples of Strategy
5a	Offer gameplay advice	Advice directed at players. Typically, this involves basic but non-obvious strategies for succeeding at the game.	Review 179: "the big pro tip here is that just when you think you've tried absolutely everything and are ready to give up, you actually haven't tried absolutely everything and need to try again"
5b	Offer game design advice	Advice directed at the game's designers or publishers. Typically either a recommendation of a new feature or a different production schedule.	Review 165: "1. A less annoying way to place pipes2. [sic] A slide control to see what exactly went wrong3. [sic] maybe the possibility to add/remove a waldo to gain a better score.(this is not really a - point but rather something I'd find neat)"

Strategy 5b showed no significant variation across site type or across individual sites. This was, in part, due to the fact that it was a rare strategy overall. Regardless, there was a slight concentration of the strategy on reviews on GameSpot. The importance of this strategy was that it addressed an audience not the one typically conceptualized for reviews: the game maker. In offering game design advice, reviewers acknowledged that audiences other than potential buyers would read the

²⁰ The one type of recommendation that I did not code into this move was the purchase recommendation as this type of statement seemed to generally be a claim about the overall quality of a game rather than an instruction for acting with the game in some particular way.

review. This, likewise, flipped the end purpose of the review: rather than evaluating the game for potential buyers, the review became a place to evaluate the game for designers in the hopes that it could be improved. We could attribute this change to a number of different forces: games are now commonly updated and changed after release and, in some cases, developers do actually respond to reviews.²¹

Strategy 5a showed a non-random concentration on two sites: GOG and Giant Bomb. In general, there was also a concentration of this strategy in the less forgiving and more complex games, as 12 of the 15 instances of the strategy were concentrated in 3 of the game genres: economy management, rogue-like, and survival. The advice ranged from general statements of the appropriate mindset for the game (e.g., advice on not giving up when it gets frustrating) to specific advice on particular actions to take in game (e.g., what buttons to press or what to pay attention to at specific times). Like the previous strategy, this one acknowledged an alternative purpose behind reviews: in addition to using the genre to make a purchasing decision, players also used the genre to improve their enjoyment of the game after having purchased it.

INFLUENCE OF INDIVIDUAL GAMES ON REVIEWS

The final trend I want to discuss was how individual games (or communities around individual games) influenced the construction of reviews. Some reviewers seemed to be aware of other reviews to the point that multiple reviews on the same site followed very similar structures. Often these patterns involved otherwise rarely used strategies. My original research questions did not include an investigation of the relative impact of individual games, so I can only offer anecdotal evidence of the ways that reviews of the same game followed unusual patterns.

²¹ I'll discuss how small developers are using game reviews in the next chapter, but there have also been several famous instances where developers have publically attacked reviewers. For instance, in 2014, Digital Homicide Studios LLC posted a response video called "Reviewing the Reviewer" after Jim Sterling (a well-known game critic) posted a highly negative first impression video of their game.

One of the most striking consistencies occurred in the opening to reviews of *Binding of Isaac* on Giant Bomb. Three of the five reviews in the sample used very similar references to the developer of the game. Table 2.15 below shows the full opening move of reviews 27, 28, and 29.

Table 2.15

Opening Moves for Reviews of Binding of Isaac on Giant Bomb

Review ID	Opening Move
27	“The Binding of Isaac is the latest game from Edmund McMillen (the mind behind Super Meat Boy).”
28	“The Binding of Isaac is a downloadable rogue-like from the minds of programmer Florian Himsl, and the artist and designer behind the acclaimed Super Meat Boy, Edmund McMillen.”
29	“The Binding of Isaac is the latest creation of Edmund McMillen, the half of Team Meat primarily responsible for the art of Super Meat Boy, a connection that shows through the Isaac's art style.”

Overall, this type of contextualization (naming the developer or the developer’s past works) was uncommon in the corpus: only two additional reviews in the entire corpus had a similar opening move (one for *Cave Story* + and one for *S.T.A.L.K.E.R.*). While the shared desire to contextualize made sense in the context of the game being reviewed (McMillen was a celebrity in indie game design circles), the incredible similarity here suggested a few different possibilities: first, these authors probably read existing reviews of the game prior to posting their own review, and second, the helpfulness rating might have been swayed by very specific pieces of information. In other words, these three reviews may have been among the top 5 most helpful reviews of the game on Giant Bomb specifically because they opened with a contextualization that other readers found useful.

At the same time, this kind of “usefulness” was constrained to a very specific localization of discourse on the game. To provide some evidence for this claim, I surveyed the other sites’ reviews of *Binding of Isaac*. Three of these sites had reviews for the game (GameSpot, Metacritic,

and Steam). Then, I looked at the openings to the top 5 most helpful reviews for the game on each site. Two of these 15 reviews also mentioned the developer by name, and one review mentioned the developer's previous game. But the delivery of this contextualization differed significantly, even when it was present. And despite its similarities to *Giant Bomb*, no reviews on GameSpot opened with this kind of contextualization.

What's more reviews of several other games in the corpus showed similar types of patterns. For example, helpful reviews of *Q.U.B.E.* on GameSpot almost always contextualized their statements by disagreeing with other reviews of the game. And reviews of *Portal 2* on Steam had a propensity for stating the quality of the game through a one-liner kind of joke. Ultimately these patterns suggest that early, popular reviews of a game might have an influence on the genre construction of later reviews on the same site.

SUMMARY OF VARIATION

In summary, there was significant variation across both site types and across individual sites. While there was still a reasonably consistent generic structure to most reviews, there was some variation in even the frequency of the most common moves.

The one move that was consistent across almost every review was the overall statement of quality. This kind of general evaluation was necessary for reviews to achieve the purpose of providing a perspective on a product.

The two conventional moves, however, showed variation that was connected to the review site. Ultimately, some of this variation seems to have been driven by the sites' review requirements. Since both Desura and Steam had no limited length requirement, reviewers were able to post shorter reviews (with fewer moves) than the other sites. On the other hand, both of these sites also had some longer reviews, which suggested that the high ratings of many short reviews was actually

a reflection of the community's preferences on the construction of the genre. This variation was the most significant difference between reviews on digital distribution and advertising revenue websites: the digital distribution sites were more likely to have a large number of short reviews, while the advertising revenue sites tended to have fewer, but more developed reviews. This variation was probably tied to how users interacted with the sites. Users of digital distribution platforms were likely using the platform immediately before or after playing a game, so many of their reviews were immediate reactions to their play experiences. On the other hand, the reviewers on advertising revenue sites had to seek out those places in order to contribute a review, making it likely that their reviews were given more forethought.

As might be expected, there was more variation in the use of specific strategies. Often, it appeared that the exigencies surrounding specific games or genres influenced the strategies employed by reviewers. Still, the study was designed to largely control for this kind of variation by comparing similar games across the sites, and there were several instances of significant variation despite this control. This suggested that certain communities had a higher interest in particular aspects of games.

INFLUENCES ON A CHANGING GENRE

To wrap up the discussion, I want to offer some thoughts on why this variation is important. To do so, we'll turn back to the example review that started this chapter:

Chased a guy with a rock, with a rock, around a rock. Game Rocks. (Sparrow_Hawk, 18 January 2014)

Just a quick jog of the memory: this review was for a game called *Rust*. It was posted to Steam, where it was listed as the most helpful review of the game. In the whole 180 review corpus for this study, this review received the second most positive helpful ratings (15,303) of any review. But this review was also representative of something else—it showed a genre in the midst of change,

a genre that did not meet traditional expectations.

And this change was controversial: 177 comments were attached to the review, many of which were a variation of the statement below.

It's the top rated review because most gamers are complete idiots who don't have enough basic reading comprehension tell the difference between 'helpful' and 'humorous.' (Miss Nuro "Fishfox," 2014)

Incredibly, the author of the original responded to these (occasionally aggressive) criticisms of his review by offering a measured explanation of the purpose and value of his review. He responded to the Miss Nuro "Fishfox" by saying:

I think there's [sic] room for things to be both 'helpful' and 'humorous'. I wrote this because it's [sic] one of the first things I did in the game, and it was brilliant! It was one of the most tense and funny experiences I've [sic] had in an online game and nearly everyone who plays Rust will or has experienced that same moment. / I think that's [sic] why this has gotten rated so highly, not cause it's a silly sentence but because most people can relate to it. (Sparrow_Hawk, 2014)

Something important was happening here, something with higher stakes than a simple difference of opinion over the quality of a review. Instead, the debate was essentially one of genre change, of what Spinuzzi calls "centripetal" forces (attempts to condense and formalize genre features) and "centrifugal" forces (attempts to innovate and change the genre). In effect, some users (represented here by Miss Nuro "Fishfox") were trying to enforce the existing and established understanding of the genre and its use within the community, while other users (such as Sparrow_Hawk) were experimenting with the genre. This contentious debate over the appropriate way to write and use a review wasn't limited to comment threads on Steam reviews: rather, it was a popular topic across a range of gaming websites in 2014 (see, for instance, Dom, 2014; Grayson, 2014; Livingston, 2014; "Reviews that try to be funny must stop," n.d.). However, this genre change was also localized to the Steam platform. While the community on Desura shared a propensity towards short reviews with Steam, they did not favor the same style as Steam. Instead, the shortest reviews

on Desura tended strongly toward general statements of a game's quality (see, for instance, review 135 quoted above). The genre was changing in contentious ways, and it was also responding to the exigencies of a specific community and space; it was responding to Steam, to the other reviews written there, to the way that reviews were read and written there, to a localized, emerging genre variant.

But the story didn't stop with discussion. Instead, as these reviews expanded, there were responses on the level of user interface. In January 2015, Steam installed a second rating system: now, instead of just rating a review as helpful or unhelpful, readers could also rate it as "Funny." Responses to this move were mixed. Some saw this as a positive reflection of how the genre was being used by the community (i.e., that the new feature would let users quickly find humorous reviews) (Valentaten, 2015), while others hoped that this could allow users to filter out reviews labeled as funny (thereby using the interface to circumvent some emerging genre expressions). Other commentators have focused on the lack of an unfunny button and the way that this omission reinforced harmful social trends in the community (Charlotte, 2015). Finally, there were also criticisms of the funny button as a technology that encouraged users to post unhelpful reviews (DarkForge Games, 2015). Ultimately, the truth value of these critiques lie in how the community chooses to use of the funny button. Is it a tag of appreciation to help others find community speech that reinforces a common identity? Is it used as a way to identify reviews that subvert the genre construction in unhelpful ways? This, in turn, will have further ramifications on the future of the genre.

But Steam wasn't the only site intentionally influencing the state of the genre. The system shared by GameSpot and Giant Bomb also made efforts to direct the construction of reviews. The following instructions were posted next to the text box where authors composed new reviews:

Don't write something dumb!

- Your review must be at least 100 words long.
- Use clear, easy-to-follow language whenever possible.
- Remember: Your words may be the words that sway a reader into buying or avoiding this product. So act responsibly!
- Remember that there's an entire game page that should already be full of story details and other facts about the product you're reviewing. We welcome full-on professional reviews, but if you're interested in keeping it short, stick to your thoughts and observations on the product.
- Don't write "reaction" reviews that exist simply to argue with other reviews. Don't assume that your readers have seen any other reviews before this one. (see, for instance, "Review DayZ," n.d.)

These instructions constrained the genre in several ways: it focused on personal reaction; it was dialogic; but most of all, it had to be smart, responsible, and "not dumb." The very drive to label some reviews as "dumb" reflected preconceptions of the genre and how it was properly used. Where the Steam funny button contained some ambiguity in its purpose, the GameSpot/Giant Bomb review instructions were a fundamentally centripetal force on the genre: certain aspects and qualities of the genre were formalized by an organizational actor both by technologies (requirements of 100 words, title, etc.) and by rhetorical influence (in the form of instructions meant to ridicule a certain form of the genre).

CONCLUSION

In the most basic form, the original hypothesis (that different gaming communities could have significant influence on the genre of video game reviews) was upheld. More than anything else, this finding was related to the emergence of specific genre variations on individual sites (e.g., short reviews, narrative reviews, memetic reviews, etc.), though there was also some variation tied to the revenue model for the site (e.g., digital distribution sites were more likely to allow short, reactive reviews).

At the same time, the study also showed a wider purpose for game reviews than the one traditionally imagined. Reviews in the corpus were directed at three very different audiences:

potential buyers of the game, game developers, and current players of the game. This last group was, perhaps, the one that is driving most of the innovation in the genre. As reviewers were speaking more and more to like-minded, experienced members of the community, the game review was becoming a place to share in-jokes, to share experiences, and to show appreciation for a product in ways that were not be wholly accessible to outsiders.

Finally, the study also showed that interface design (and technical communication) could have an impact on the genre. Steam's placement of reviews on the main page for a games likely influenced the community's use of the genre to engage in strong lexical customization. Meanwhile GameSpot and Giant Bomb's "don't write something dumb" instructions (along with their complex editing platform that allows authors to integrate images) created an environment where fewer, but longer and more complex evaluations of games were valued.

The question then becomes of the appropriate place for organizational influence on the genre. Should a site constrain centrifugal forces (i.e. innovative)? What cost do they pay for doing so (e.g., lower use rates of the genre)? Is there a certain point at which, regardless of community preference, too much variation in a review genre could negatively impact the organization hosting the reviews? Alternately, are there conditions for an organization could use to encourage genre experimentation? In part, the answers to these questions come down to how technical communication sees itself in relation to this genre (and to user generated content in a larger sense). If the field maintains its long-standing commitment to user advocacy, then it seems only right to explore the actual use of the genre and to create systems that support that use. If, however, the obligation must be to organizational stakeholders (e.g., site owners or game developers), then the obligation turns to discovering how these entities use the reviews (e.g., get site views, promote sales, collect feedback on products) and to create a system that leads the genre in that direction.

FUTURE RESEARCH

While there is significant room for further investigations of the video game review genre, investigation of the use of narrative seems like a particularly fruitful angle. The corpus for this study does not offer enough data to make any significant claims about this particular feature of reviews. However, a future study could sample from games similar to Rust where the reviews have high rates of narrative in order to explore how narrative is used in those reviews. This could be a particularly fruitful direction for future research because of the established research on both narrative in games (which drove the contentious narratology vs ludology debate) and the established interest in narrative reviews of experiential products.

Another possibility of a future study would be to explore the apparent influence of highly rated reviews on later reviews of the same product as discussed earlier in the section on the influence of individual games. Such a study would need to track review construction actively over time in order to provide appropriate longitudinal evidence.

CHAPTER 3 OPEN DEVELOPMENT OF VIDEO GAMES

In 2009, Markus “Notch” Persson released *Minecraft*. He published an alpha (incomplete, early stage of development) version on TIGSource (a game developer forum), and from the start, he worked on the game in public:

Markus updated his blog often with information about changes in *Minecraft* and his thoughts about the game’s future. He invited everyone who played the game to give him comments and suggestions for improvements. In addition to that, he released updates in accordance to the Swedish saying “often rather than good” (meaning someone who prefers spontaneity over perfection). As soon as a new function or bug-fix was in place, he made it available via his site, asking players for help in testing and improving it. (Goldberg & Larsson, 2015, pp. 98–99)

Minecraft has gone on to sell more than 22 million copies on PC (and many millions more on other consoles), and has become something of a cultural phenomenon (Makuch, 2016). Meanwhile, Open Development (OD), the methodology²² used by Notch, has also exploded in popularity. OD certainly wasn’t initiated by Notch; several previously-released, popular games also used OD methods, including *Dwarf Fortress* (Rose, 2013) and *Counter Strike* (Te, 2014). Yet the massive expansion of the OD methodology (and of related crowdfunding and Early Access programs) can be partially traced back to *Minecraft*’s success.

Broadly speaking, OD involves releasing incomplete (prototype, alpha, or beta) versions of the game to the public and gathering player feedback which helps to improve the game. While it introduces many new practices, OD shouldn’t be thought of as a rejection of traditional game development. Developers have used members of the community to playtest games for decades, though most of this testing occurred behind closed doors in the form of in-house tests (where players are invited in and made to sign non-disclosure waivers) or in closed betas (which are open only to a select group and number of people). The main change in OD is the expansion of this

²² I refer to Open Development as methodology throughout this chapter because it can be more closely associated with general principals than with a specific set of methods.

testing community to, literally, anyone who is interested. Meanwhile, many other aspects of game development remain unchanged—developers still come up with the main idea of the game; they code and design the art and sound for the game; they do internal quality assurance testing; and they publish and market the game.²³

Many developers who've undertaken OD projects have stated that the main change, for them, was in communication practices. For example, Unknown Worlds (n.d.) stated, "Truly, the practical application of open development is about communication." Similarly, while discussing *Unreal Tournament*, a recent and high profile OD game, Wawro (2015) stated, "When players become developers, developers must become communicators." And OD developers recognize that they still need to discover additional ways to communicate with players (see, for example, Avellone's discussion of communicating game development progress on p. 25 of this chapter). In developing a definition for OD then, it is essential that we attend to the activities of communication that comprise the majority of the new labor associated with the methodology.

It is also the industry's attention to communication practices that allows me to firmly situate the discussion of OD within the framework of technical communication. For years, technical communication has been seeking productive ways to respond to and build connections with the gaming industry (see Eyman, 2008; Greene & Palmer, 2014; Mason, 2013; Peterson, 2004). OD offers a frankly obvious means to pursue this goal. Given the interest of OD in communicating effectively with users/players (a lay audience), and given technical communication's long experience with usability/user-centered design/user experience (including more recent work to redefine technical communication as one part of the larger field of user

²³ Of course, all of this also remains open to changing definitions of the new methodology. Some developers are certainly beginning to press the participatory nature of OD beyond the confines of traditional game development, but this isn't necessarily a practice spread broadly across OD adopters.

experience (Redish & Barnum, 2011; Six, 2015)), technical communication is positioned well to offer theoretical and methodological rigor to the emerging practices of OD projects.

The first step in this process is to develop a clearer definition and understanding of the methodology. This chapter will pursue this goal through two methods: first, it will narrow the definition of open development through a series of activity theory analyses, then it will take a closer look at how this activity is mediated by mapping the genres associated with work in the OD methodology. Thus the primary argument of this chapter is that these activity and genre mapping methods can uncover the underlying constraints and contradictions of the methodology, and, in doing so, can help to create a foundation for improved practice. Specifically, the activity theory analysis will show how contradictions in OD activities have contributed to failures in past OD projects. The genre analysis further specifies the ways in which OD activities have been carried out, and also help to construct a way to address current breakdowns in OD activity.

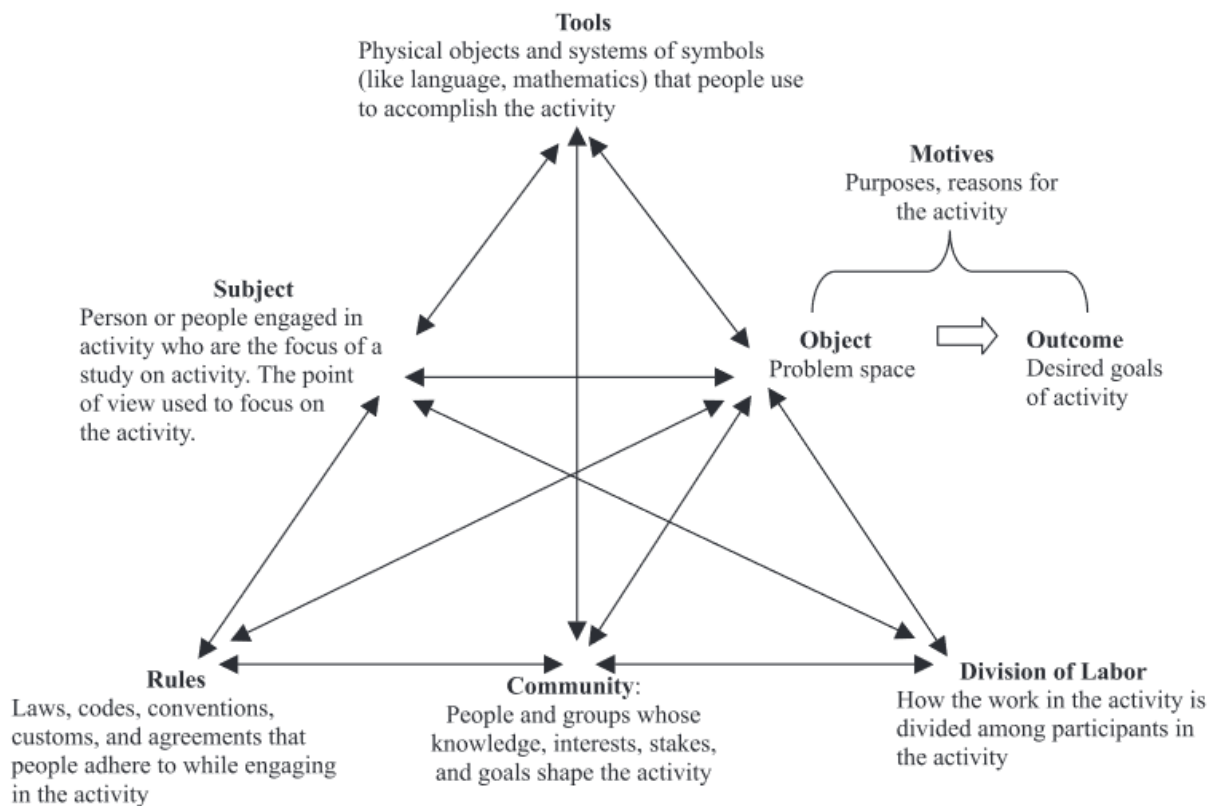
ACTIVITY THEORY ANALYSIS

Activity theory and genre analysis are fundamentally intertwined. Activity acts as the macro level of analysis above genre or as the broad situation in which any genre system is embedded. Activity theory grew out of the socio-cultural tradition of Russian psychology through the work of Vygotsky and Leontiev, who forwarded the concept of mediation. In short, mediation showed that subjects achieve the object of their activity only through the mediation of tools. These tools were defined broadly in order to include anything from physical tools (e.g., hammers and keyboards) to linguistic tools and conceptual tools (e.g., languages and genres). These tools did not necessarily act unidirectionally (i.e., gaining agency from subjects in order to impact objects), but instead also became part of the internalized processes that governed how a subject would be able to understand an activity (Kaptelinin, n.d.). Most important for the analysis of OD activity

here was Engeström's (1987, 2001) graphical representation of CHAT (Cultural Historical Activity Theory), which detailed the underlying cultural aspects of the mediated subject-object relationship as seen in Figure 3.1 below. Notably, CHAT relies on three sets of mediations (tools, rules, and division of labor) through which the subject of an activity impacts the object of the activity.

Figure 3.1

Activity Theory Diagram (Kain & Wardle, 2005, p. 120)



Several scholars have already discussed this diagram in significant detail (see, for instance, (Engeström, 1987, 2001; Kain & Wardle, 2005; Kaptelinin, n.d.), so instead of replicating this work, we might clarify the concepts through Nardi's (1996) example of the activity of a software development team:

The object is the not-yet-ready system, which should be transformed into a delivered, bug-free application. The team is the community sharing the object, perhaps joined by

some representatives of the customer. There is a certain division of labor: between manager and subordinates, between software developers and user representatives, and between the team members. There is a set of rules covering what it means to be a member of this community. Some of these rules may be explicit--set by law, the parent organization, or the team manager--but many of them are most certainly implicit, either as a part of the general working culture or developed as the team works together. Some rules may be constructed for this particular project ... In each step of the transformation process, a different set of tools and instruments is used in the transformation process--for example, analysis methods, computers, programming tools, walk-throughs, or rules of thumb. ... Whatever members of the team do during the project is shaped by the context of activity. (p. 29)

This example showed how activity could be used to sketch the outlines of actual processes in ways that help to enumerate the underlying social structures that influence the formulations of subjects and objects. In much the same way, the following sections will take up an analysis of OD activity in order to better depict the labor and goals of those involved.

THE ACTIVITY OF OPEN DEVELOPMENT

In this first analysis, I want to start by approaching OD holistically. Before jumping into the analysis of these specific elements, we can briefly return to definitions of OD in order to expand our initial understanding of the methodology. While there still isn't a universally accepted definition, several developers have offered their own take on the methodology:

- “Open Development means building a game completely in the open, where every aspect of development is exposed to the public, where every change affects the game in real time, and where anyone can contribute to the process.” (Brown, 2016)
- “‘Open game development’ is an oft quoted but somewhat nebulous maxim of Unknown Worlds, and many other indie game developers. Perhaps it can be defined as: ‘Game development in which the decisions, processes and people inside the developer are visible to consumers and other outside parties.’” (Unknown Worlds, n.d.)
- “Creating a more transparent development process while profiting from the community

experience – that is our idea of open development.” (Winter & Mroz, 2014)

While these definitions varied slightly, there were a number of underlying similarities. First was the focus on visibility, transparency, or exposure. Developers primarily connected the idea openness with visibility. Likewise, the developers discussed the audience of this visibility as the public, consumers, and/or the community—these three concepts had slightly different inflections (community presumably focuses more on players while consumers explicitly focuses on potential players), but they all signaled a broad audience outside of the development team. Finally, the topic of this visibility was described broadly as the development process. More than anything else, then, open development was conceptualized as a process of making typically non-public elements of the development process visible to people outside of the game development organization. However, these definitions did not explicitly talk about method—they did not signal how this information is made visible to the public, nor did they reference systems by which the public (or players) communicate with the developers. Still I would argue that this latter aspect (of feedback) has been essential to the functioning of OD *as* development rather than as simply marketing. In the following sections, I will describe each element of the activity system by referencing discussions of OD. After each of element is described, I will then offer an overall summary of the system in an activity diagram.

Subject and community. The definitions above suggested that the dev team was the most likely subject (or focus) for an activity theory analysis of OD. However, developers have also spoken of OD as a much more collaborative activity:

- “Open development is about exploring your design with players as you build your game.”
(Luck & Day, 2015)
- “We want the player community's help to 'evolve' ARK into the dinosaur world game that

we've been dreaming about.” (“ARK: Survival Evolved,” n.d.)

- “When we talk about community development, it is really about having the community and the dev team exchange design and gameplay ideas as a direct feedback loop, from the community, right into the dev team.” (Spock, 2014)
- “We like to think of games and game development as services that grow and evolve with the involvement of customers and the community.” (“Introducing Early Access,” n.d.)
- “You can only say that you are part of your community if you share something with them, if you talk with them, if you answer questions and ask questions yourself. And it's not ‘you & the community,’ it's everybody in the community, including you.” (Castelnerac, 2014)

Notably, these statements suggested that a certain level of identification between the developer and players was necessary for a successful OD project. Speaking holistically, we could then define the subject as the combination of the developers and the players. This perspective would acknowledge the varied contributions by both groups (rather than prioritizing the work of only one).

However, it would also limit the potential definition of the community (i.e., if the definition of the subject is so broad, who is left to make up the community?). Returning to Figure 1 above, we can see that the community can be thought of as anyone whose “knowledge, interests, stakes, and goals shape the activity” (Kain & Wardle, 2005, p. 120). Based on this definition, the community would include other potential influences, including the publisher, consumers, games journalism, and the broader gaming community. In each case, these groups could influence the outcome of OD activity without being the primary originator of most actions within the project.

Object and outcome. Rather than turning to discussions of OD for these qualities, we can draw on Nardi’s example of software development activity above, where the object and outcome

were the “not-yet-ready system” and “a delivered, bug-free application” respectively. In much the same way, the object and outcome of OD, as a game development methodology, is the current, incomplete iteration of a game and a final release of a complete, good game, respectively.

Motive. In activity theory, the motive is subjective or social reason that drives the movement from the object toward the desired outcome. In our current analysis, this must account for why OD is used to move a prototype toward a finished product. Several developers have commented on why they choose to use the OD methodology:

- “It helps you design better games.” (Luck & Day, 2015)
- “At Unknown Worlds, we like making games in the open. It's more fun that way, and we think it makes better games.” (“Subnautica,” n.d.)
- “We like to support and encourage developers who want to ship early, involve customers, and build lasting relationships that help everyone make better games.” (“Introducing Early Access,” n.d.)

The concept of “better games” became a refrain throughout these statements, but the exact meaning of “better” was not specified (better for who? better how?). Still we can take their belief in the improvement of games as the developers’ motive for engaging in the OD methodology instead of more traditional development methodologies.

Tools. With the rest of the activity system outlined, we can turn to the three mediators of activity, namely tools, rules, and division of labor. The discussion of tools in OD has tended towards specifics of what was being used in individual game projects. A basic list could include update announcements (Johnson et al., 2015), forums (Castelnerac, 2014), blogs (Winter & Mroz, 2014), a development schedule (with time built in for responding to community requests) (Crowe, 2014; Spock, 2014), a list of proposed/upcoming changes (Castelnerac, 2014), surveys/polls

(Spock, 2014), development videos (Garriott, 2014), and player-built content (Garriott, 2014).²⁴ Broadly speaking, these tools can be categorized as developer-player communication or as game development/project management tools. The one exception might be the player-built content, but this tool isn't even used in every OD project.

Rules. Like many things involved with OD, we can first assume that the standard rules for game development remain the same (e.g., that a game should be fun, playable, and usable in a broad sense). Drawing on the definitions of OD above, we can add two additional rules that apply specifically to OD projects: namely transparency and collaboration. OD projects have generally relied on these two rules to structure interactions between developers and players with the goal of creating a better end product.

Division of labor. We can start by assuming a fairly traditional division of labor within the dev team along game development sub specialties (e.g., project management, programming, level design, writer, art, sound, quality assurance, marketing, etc.) (Liming & Vilorio, 2011). From there, we might also add in community management, which has been alternatively handled by the whole dev team (Brown, 2016), by specialists within the dev team (Castelnerac, 2014), or by representatives within the player community (Garriott, 2014). Outside of the dev team, we can also find some division of labor. First, many player communities have been divided among active participants and “passive” observers (Garriott, 2014). However, I want to emphasize that, as Beller argued (2006), even the observers were producing labor: watching open development created an increased network effect (an increase in the discoverability of a project based on the number of individuals observing it) and created subjectivities amenable to open development media. In some projects, community members have also acted as content-creators or modders (Brown, 2016;

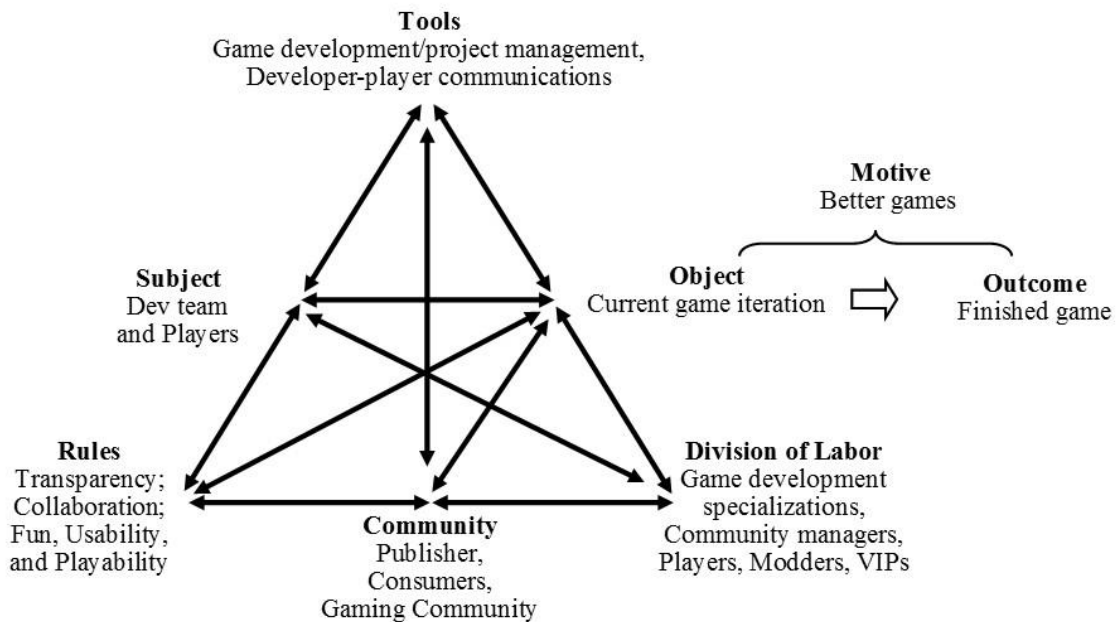
²⁴ I discuss these tools in much more detail in the section on rhetorical genres used in OD activity.

Garriott, 2014). Finally, some projects involved player community VIPs, who were brought in at an earlier stage of development (often pre-development or prototype) than the rest of the community, and who had a greater influence on and a more direct access to the dev team (Spock, 2014; Winter & Mroz, 2014). Using these concepts we can begin to divide the different types of work that the dev team and players engaged in.

O.D. activity conclusion. We can summarize the discussion above in Figure 3.2 below.

Figure 3.2

Open Development as a Unified Activity



The above analysis had the advantage of offering a high level view of OD activity as a whole. However, this conception of OD risked oversimplifying and idealizing the complex interactions that have characterized the specifics of actual OD projects, which have not been nearly as harmonious as this depiction might suggest. Part of the issue was that this description was built solely on developers' public descriptions of their work: "And while many developers are positive in their assessment of the model in the public, some have complained in private of extremely

demanding users, contradicting feedback, and controversies that blow up and threaten to scuttle their game before it's even complete” (Bailey, 2015). Likewise, player descriptions of their experience with open development games weren't always positive:

- “90% of these games never make to any sort of "release quality" experience, and by that I mean they're unfinished, bug riddled, crap, even though "we" paid to alpha/beta test them. These devs just take their money, make it into the black and then realize where it's time to cut tail and run.” (Mr. Dingleberry, 2013)
- “The player gets an unfinished game for their money with a promise of finished game in the future.” (so_hai, 2015)
- “A game can be ruined by Early Access before it's ever released.” (FoxeoGames, 2015)
- “Early access means a game is less fun to both read about and play.” (Plafke, 2013)

Of course, not all player accounts of OD were this negative; however, the presence of these varied and negative perspectives suggested that the unified view of OD activity was oversimplified and that an alternative model needed to be constructed to allow for further differentiation and conflict between perspectives. Fortunately, activity theory can also help us to think through this differentiation. For example, Nardi (1996) expanded on the previously cited example of software development activity by stating:

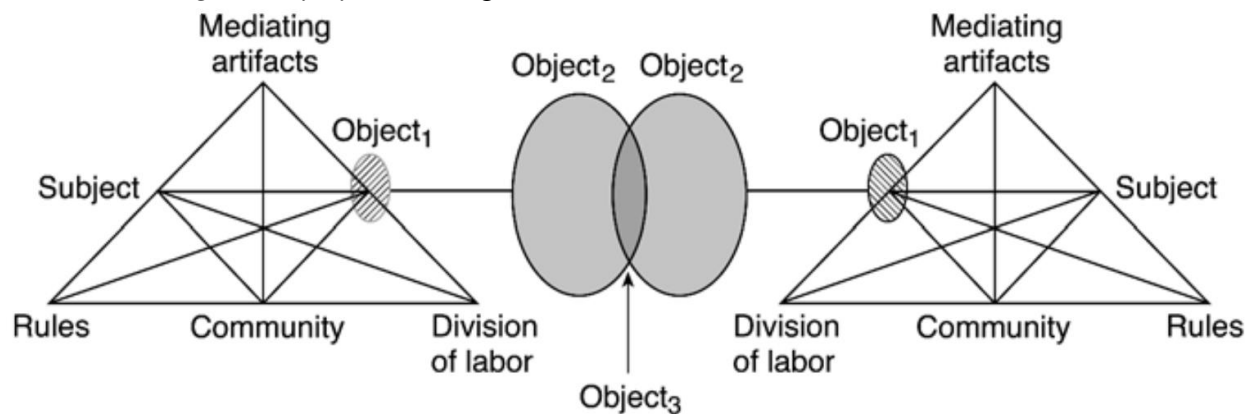
At the same time there is another activity; here the object is the financial status of the software company, and the community consists of team managers and their superiors. Every team manager has tools and tricks designed to keep the project within budget and profitable, and superiors have their own as well. There is a certain division of labor and a certain set of rules--most certainly different from that within a team. / We can imagine a third activity whereby some of the team managers compete against each other for an available position as a department manager. (29)

In short, activity theory allows us to look at differing (and sometimes contradictory) perspectives and practices that interact and contribute to a broader conception of the activity in

question. Likewise Engeström (2001) stated that “The third generation of activity theory needs to develop conceptual tools to understand dialogue, multiple perspectives, and networks of interacting activity systems. ... In this mode of research, the basic model is expanded to include minimally two interacting activity systems” (pp. 135-136). He modeled this concept of activity theory with the diagram show in Figure 3.3 below.

Figure 3.3

Two Interacting Activity Systems (Engeström, 2001)



The following sections will take up this multiple system perspective in offering a further analysis of OD.

OD ACTIVITIES

In this section, I will outline three activities that have been fundamental to the rhetorical functioning of OD. In doing so, I first want to return to the concept of OD activity as being differentiated from traditional development in terms of communication activity. For this reason, this analysis will focus on three communication activities, namely game design communication, marketing communication, and player communication. Outlining these three activities will help us to see the contradictions inherent in holistic OD activity more broadly.

Design communication activity. While Figure 2 above represents a holistic view of OD as game development, I want to re-formulate the developer perspective here as one inherently

involved with the communication of design to player partners. This activity broadly encompasses several types of work, including the initial representation of a design vision to players, the subsequent negotiation, revision, and defense of that design visions, and the continual updating of the development progress—put together this work represents the way that developers use OD to create and advance the design of their games.

With the subject of the activity limited to the dev team, the players become the focus of the community. While there were other, secondary audiences for much of the design communication activity (games journalists, consumers, etc.), they often didn't take an active role in these communications, so we can set them aside for now.

With the shift toward communication activity, the object and outcome could no longer be seen as the current and final game iterations, but rather the game design itself (with the object being the current design and the outcome being the design at the time of the official release). In this way, the design acts more as an object (as something to be worked on) than as a concept (as something that exists inside our minds only). It is created not only through the traditional negotiations between members of the dev team (and through the technological mediations of development software), but also through communication with the player community.

We can see this even more clearly if we look for the OD dev teams' motives for engaging in communication with the community. For example, Luck and Day (2015) discussed how they observed the community playing the game and used it to weed out overpowered strategies, resulting in a more balanced game. S. Johnson et al. (2015) described how they created updates for a game that would test out different trajectories and then use player feedback to evaluate that path. And Brown (2016) discussed how player feedback could “let a developer know exactly what they want from a game, rather than relying on the designer's assumptions or interpretations.” Put

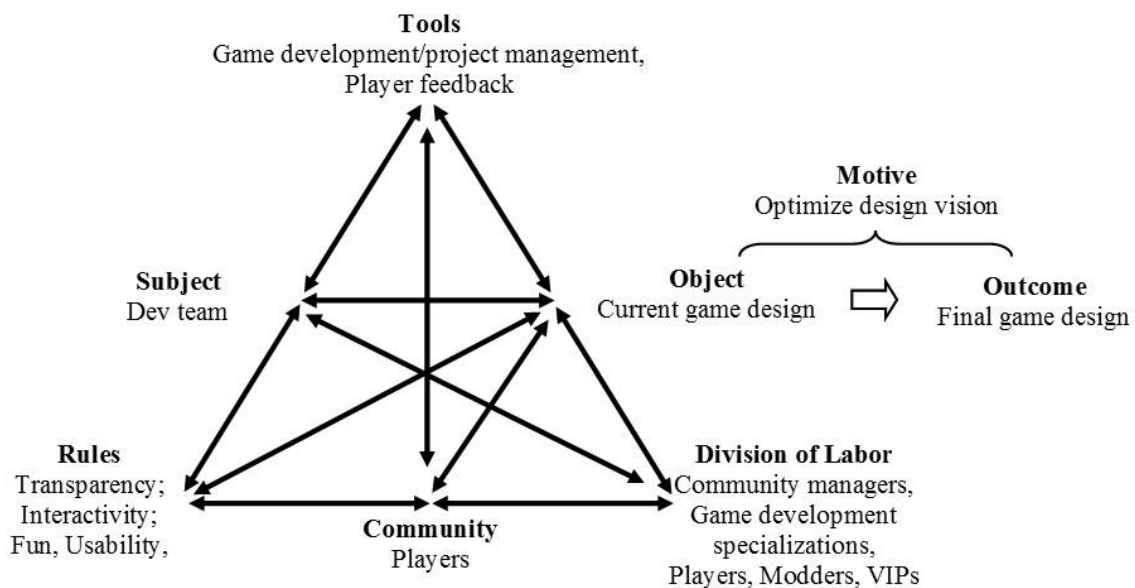
together, these statements suggested that developers valued OD interaction both as a laboratory to test out ideas and as a means to gather real information about player wants. None of this meant that OD was, as Walker (2014) argued, crowdsourced design and pandering to the lowest common denominator of the gaming audience. In fact, many developers have repeatedly emphasized the ways in which OD dev teams remained in charge of and responsible to their own design decisions (Castelnerac, 2014; Luck & Day, 2015; Roth, 2014; Spock, 2014). Likewise, Sigman noted that “You still require your own map and compass during the process, and sometimes you need to hold course based upon your own intuition instead of the incredibly high volume of (often conflicting) feedback” (qtd. in Bailey, 2015). In these cases, it was the onus for developers to accurately and persuasively describe their design decisions (rather than simply giving into vocal members of the player community). Yet many developers seemed to find this process of explaining their design to be a positive experience that helped them to refine their initial design (Johnson et al., 2015). Thus, we can see the activity of developer driven design communication as one fundamentally caught up in the rhetoric of bringing players into a common understanding of what the ultimate game design ought to be, while also folding in particularly useful player communications into a revised and improved version of the design. In this way, the motive of this activity was to optimize the game’s design (which, from the developer’s perspective, involved the realization and fine-tuning of their design vision).

Turning finally to the mediating aspects of this activity theory analysis, we can leave most of them unchanged. The division of labor encompassed the work of the different elements of the dev team, though the community manager would take on additional prevalence in this analysis. The rules revolved around transparency and good game design, though this focus involves something less than collaboration that we might term as interactivity (in that the developers

interacted with the community to fine tune the game design). The tools also still involved project management and game development practices and tools, but the developer-player communication narrowed simply to player feedback as a tool that directly impacted the dev team's ability to optimize their design. The resulting activity theory of game design OD communication is represented in Figure 3.4 below.

Figure 3.4

Open Development as Design Communication Activity



Marketing activity. Many people have rightly pointed out that OD is not just a design activity, but also a marketing activity (Castelnerac, 2014; Crowe, 2014; Lindskog, 2014; Luck & Day, 2015). From the start, OD has been used as a way to create funding and to establish a stronger market presence. Notch notably used crowdfunding to support *Minecraft*: “At home in Sollentuna, Markus did a quick calculation: *If I can sell more than twenty games a day, that’s enough for something approaching a decent salary*, he thought, and made up his mind. *Then I’ll quit my day job. Then I’m really doing this*” (Goldberg & Larsson, 2015, p. 101). Likewise, Wolfire, an early

adopter of OD for their game *Overgrowth*, described OD as primarily a PR strategy that could help small, independent developers “make noise” or draw attention in an overly saturated market (Graham, 2009). One response to this marketing activity has been to differentiate OD as development from more financially-focused aspects: this impulse led to differentiating between OD and Early access (Luck & Day, 2015) and between OD and crowdfunding sites such as Kickstarter (Garriott, 2014). Still the hard division implied in these arguments overlooked how both marketing and design communications were often present in the same places, and how some rhetorical genres served both ends at once. Instead of fully dividing the activities then, it would be productive to see both as integral elements of a larger OD activity. In the next several paragraphs, I will outline the activity of OD marketing. While it was tempting to use marketing scholarship to construct this description, the field has only just begun to explore the potential of activity theory as a systems-based approach to studying their work (Ng et al., 2012). With that in mind, I will mainly refer to OD scholarship below, but will occasionally bring in marketing scholarship in order to complicate some of the depictions.

As a marketing activity, OD could be described through a focus on a publisher. In traditional game development systems, publishers have often been separate from developers, but since most OD games have been produced by small companies, and since Early Access systems have allowed for self-publishing, the OD developer and OD publisher were typically the same company. The community here was primarily composed of consumers (who have yet to purchase the game, but who engage in production through attention), joined by games journalists a major secondary audience (Wolfire’s concept of making noise explicitly focused on having games journalism sites pick up on stories related to their game). Finally, players also contributed to the work of community through writing player game reviews, which acted as free advertising for a

developer.

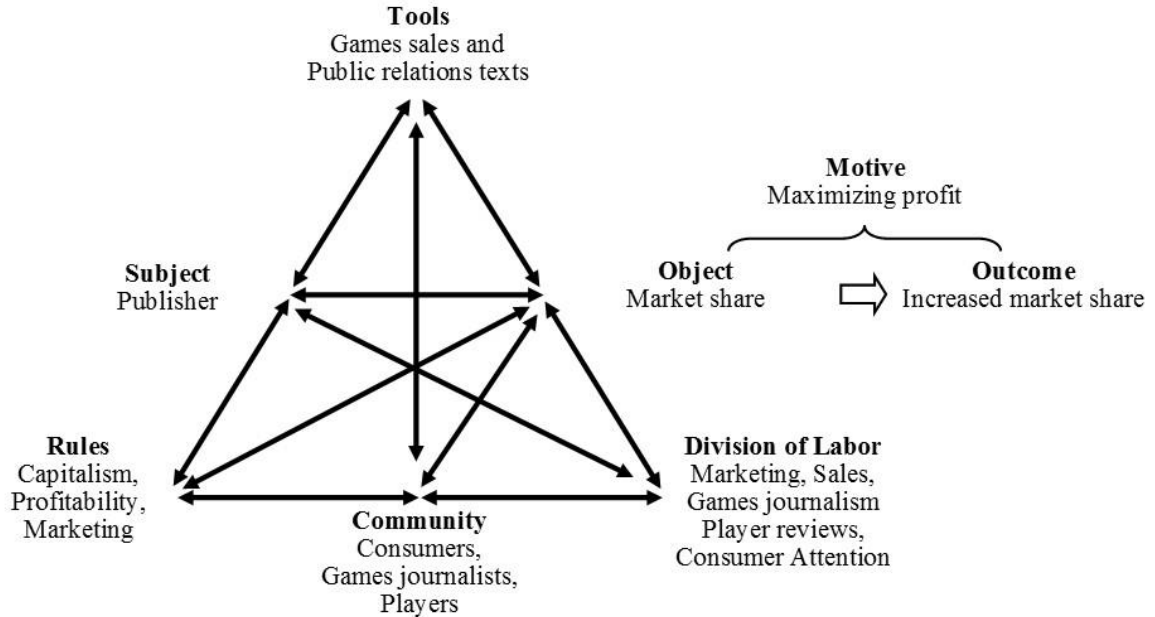
The object and outcome were a bit more ambiguous here, and we might initially conceptualize them in any of a number of ways. First, income (or sales) might be seen as the object, with an outcome of maximizing profit. However, this seems to be contradicted by marketing scholarship, which has emphasized a broader view of the process: for example, McKenna (1991) argued that “The real goal of marketing is to own the market—not just to make or sell products.” Likewise, consumers might be seen as the object, with the outcome of maximizing the consumer base. Castelneac (2014) adopted this perspective when he said that “open development enables more and more people to understand what you are doing.” This perspective would help to show marketing as acting directly on an object (the consumer), but it also might represent an oversimplification of both the functioning of rhetorical communication and the role of consumers. Finally, we could also describe the object and outcome in more nebulous terms, like Wolfire’s concept of “making noise.” In order to capture the greater range of the latter concept while also retaining something a bit more concrete, we can identify the object and outcome as current market share and an increased market share. The motive for capturing market share then can shift to profit (or at the very least solvency) so that developers can continue their work.

The tools still focused on communication, but shifted focus toward consumer-oriented texts such as game distribution sites and PR genres, such as social media communications. Basically, the tools included both the means of selling the game and the means of publicizing information about the game and about the development process as fundamentally engaged with players. The rules shifted away from game design and towards capitalism, profitability, and marketing. And finally, the division of labor acknowledged the contributions from secondary community members (journalists and players) as noted above. An overview of OD marketing activity is represented in

Figure 3.5 below.

Figure 3.5

Open Development as a Marketing Activity



Feedback Activity. Shifting the subject focus toward the player was primarily difficult because of the dearth of authoritative accounts of this aspect of OD. What little was available could primarily be culled from forums, reviews, reddit topics, and comments sections of articles; however, given the genre characteristics of these communications, most player accounts found here were brief and often contradictory. Longer accounts of the player perspective were usually written by games journalists, who were channeling their role as representatives of the larger gaming community. While I drew on their views here as well, journalists sometimes have different goals than normal players: for example, as McAllister (2004) argued, games journalists can have a vested interest in supporting developers (p. 133). All of this is to say that the player-centric view of OD that I developed here was based on more esoteric and variable evidence than the developer-centric activities described above. Even with this limitation, it was still essential to speak to the

feedback activity of players in order to capture the affordances, limitations, and (perhaps) conflicting goals that characterized OD as a complete activity.

With the player firmly in the subject position, the community was primarily composed of the dev team and other players. (The dev team can play an especially significant role in constructing the environment for feedback.) While it was initially tempting to retain the object and outcome of the game design activity, this identification would have glossed over the fact that players do not work directly on the design. Alternatively, players might have been described as working on the dev team, though as with the previous discussion of marketing activity, this would posit a too direct (and unmediated) communicative relationship. Instead, I turned to player discussions (from comment sections and forums) to see what they said they were working on:

- “For the honest developers out there, [early access] provides feedback, support and a player base which in turn helps them polish and adjust their games.” (johnd13, 2015)
- “I try to help devs where I can with positive and constructive feedback.” (Hobbes, 2015)
- “What I also enjoy about the early access state is that I can in my own way, help craft the actual game with suggestions and bug feedback as the game is developed.” (blackknight115, 2015)
- “If you are passionate about a project, and want to help guide a games' progress and make suggestions, and report bugs and other problems in a constructive way, then EA is great.” (GrahamZ, 2015)

Players conceptualized their activity as having the object of “feedback” or “suggestions.” This object contributed to the outcome changes to the actual game. This object and outcome reflected a self-described player role that was far more limited than idealistic collaboration forwarded by developers—players saw themselves as just involved in the work of feedback rather

than in some grander process of design work. Given this limitation, it was harder to establish the motive behind players' enthusiasm for engaging in these projects. Again, I turned to player commentaries further specify the motive in feedback activity:

- “It’s awesome. We’ve seen a lot of games get made recently that could’ve only been done via kickstarting and crowd funding.” (KHAndAnime, 2015)
- “I love it...great way for both smaller developers to get ideas that normally wouldn't get published out there, and for gamers to get their hands on earlier versions of the game to help shape what they might become.” (Revenent42, 2015)
- “It’s your voice, taken a step beyond the expression of interest you can make through the Greenlight program by giving you a concrete stake in what you claim to want. When you see a game idea you love being handled by people you trust, put your money behind it, and help make that experience you want a reality.” (Wiesehan, 2014)
- “If you are passionate about a project, and want to help guide a games' progress and make suggestions, and report bugs and other problems in a constructive way, then EA is great.” (GrahamZ, 2015)

A few threads ran through these comments, but most common was the desire to see certain types of games—basically, these players expressed an interest in influencing the types of games being built. Therefore, the motive for engaging in feedback activity was then one of fit—players tried to persuade the developers to change the game in a way that moved it toward their desired experience (whether this be an experimental gameplay type that would not have been published otherwise, a more stable game build or a game with a specific focus or features that had not been planned). In other words, we can understand the fundamental activity of feedback as one of rhetorical realignment in which players sought to persuade developers to adopt a particular vision of design

(or development priorities).

Of the mediatory elements of this activity analysis, the tools and division of labor were relatively clear while the rules were less certain. The tools were comprised of genres of feedback and channels for communication. Together these helped the players know how to construct feedback and how to deliver that feedback to developers. The division of labor was described in the earlier discussions of VIPs, community leaders, participators, and watchers. These roles were often fluid within communities, but they reflected differing levels of commitment that came with the potential for different actions (for example, some VIPs had substantial, direct conversations with members of the dev team, while some general participants could only vote occasionally in developer-created polls). The rules were less well-defined, partially because OD player communities varied and partially because individuals within each community participated in different ways. While there were not many places where these rules were stated explicitly, some developers did establish guidelines for providing feedback in sticky posts on their forums. These posts contained three main themes:

- **Active Participation:** Players were encouraged to give feedback and were linked to appropriate communication channels on which to do so (BitCake Studio, 2016; Linds, 2016). This rule effectively said that if players wanted to see a change, then they had to make a suggestion. Sometimes this encouragement was tempered with the request that players look at existing forum threads/feedback before contributing (Fiset, 2016).
- **Constructive Feedback:** Developers often asked specifically for “constructive” feedback, though they never defined the meaning of the term. For example, “Be constructive and respectful with criticism and feedback” (BigPun, 2016), and “Make sure you are posting constructive criticism, whether it's negative or positive” (SeeMeScrollin, 2016).

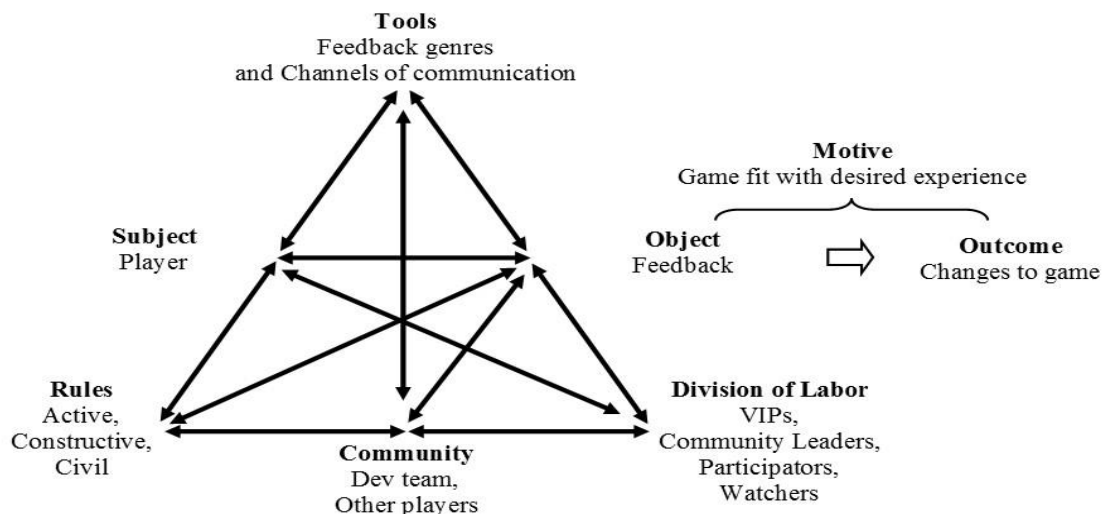
- Civil discourse: Several forum sticky posts emphasized etiquette in interactions between players and developers as well as amongst players themselves. In many cases, the primary rule was that abusive or threatening behavior was not allowed (alex_sawczuk, 2016). These guidelines were also sometimes built explicitly on Steam's rules for community etiquette (BigPun, 2016; SeeMeScrollin, 2016).

Of course, it was difficult to determine the extent to which these rules were actually operationalized by players engaging in feedback activity, as numerous developers have described the experience of receiving torrents of angry and disruptive feedback (Castelnerac, 2014; Hall, 2015; Johnson et al., 2015) and reviews and discussions on Steam still display some of the vitriol that is altogether too common in online discussions of gaming (for example, after a recent controversial change to the game, popular review of *HIZI* depicted an ASCII text art image of a person defecating on the game).

Combining all of the above elements, a summary of OD feedback activity is represented in Figure 3.6 below.

Figure 3.6

Open Development as Feedback Activity



Combined O.D. activity. Further activities that influenced open development games could be mapped out: for example, community building player to player communications have had a significant impact on the longevity of several games (RPS, 2015; Sullivan, 2015). Yet, the three activities described above had the most direct impact on how OD games were designed and developed, thus making them the most integral elements in the larger OD system. We can (simplify and) further map these three activities together to depict OD as a whole as shown in Figure 3.7 below.

Figure 3.7

Combined O.D. Activity



By dividing OD activity in this way, we can begin to see some complications that were invisible in the unified view of OD activity presented in figure 2. To begin with, by viewing OD as the convergence of multiple activities, we can begin to conceptualize the breakdowns in the system that arise from contradictions between the objects and motive of the individual constituent activities. The first breakdowns occurred when game fit became prioritized. Walker (2014)

described fit overriding design when he argued that OD projects pandered to the least common denominator and were only able to replicate features and systems that players had already experienced. Likewise, when fit was prioritized over profit, developers became engaged in never-ending cycles of bug fixes and changes to make the community happy without making significant progress toward completion. In a similar cases, when design was prioritized over profit, games also entered a perpetual development phase where grandiose designs were pursued without any plan for completion. In either case, these breakdowns led to seemingly slow development cycles further aggravated tensions between players and developers. For example, Cooper (2016) found that only 25% of Early Access games had been released. Likewise, fans have complained that several high profile OD projects felt like they would never be finished (for example, see hillkill's (2015) assertion that DayZ will never be done). Many people have already argued that this complaint about perpetual development is likely tied to players' unrealistic understanding of/expectations for development speed, but as Coghlan (2015) argued, this breaking of expectations still had a significant and negative impact on the perceived results of OD projects. Likewise, expectations for player participation were broken when design overpowered fit. In these cases, players complained about being ignored by the developer, about developer silence (see Crawley, 2014), or about games incorporating undesirable features, such as Godus' freemium model (Grayson, 2015). Finally, the most prominent complaint about OD games, that they were a "scam," was tied to situations where profit was prioritized over either design or fit. For example, *Spacebase DF-9* was abandoned by its developer because it failed to meet sales expectations, causing significant backlash by players who felt that promises were broken (Maiberg, 2104). A less egregious example of marketing overriding design communication occurred when the authenticity of the OD transparency was questioned. Winter and Mroz (2014) discussed this how

this sort of dilemma forced them to limit some of their communications:

We as developers want to give the fans funny and spontaneous information, which probably isn't too marketing-compliant. But the marketing team wants to stay in control, so every post needs to be planned and preapproved, which makes it very hard to be spontaneous. ... The sacrifice of spontaneity and virality just has to be made.

All of this is to say that effective OD projects functioned through the balancing of the activities described above, while failed projects were characterized by one or more imbalances. With that being said, many developers have already begun to take steps to ensure a more harmonious balance in OD projects, from providing ongoing lists of development goals to setting hard deadlines for release of specific features. Likewise, players have collectively tried to re-orient their expectations by arguing that promises for the finished project should not be the reason for financially supporting an OD project (as opposed to an appreciation of the current version or investment in offering feedback).

In addition to giving us a high level view of OD, this activity analysis can also act as a system for categorizing work within the specific genres of OD projects. I will undertake the task of explicating the specific genres in more depth in the next section.

OD GENRES

By returning to the concept of genres that has driven the other chapters in this dissertation, we can explore OD practices more directly. My original goal in this chapter was to explore the specifics of the relationships and characteristics of the localized genres of Open Development, but as I began this process, I found the lack of a commonly accepted understanding of OD made it effectively impossible to talk about the specific functions of genres. (For example, without the framework described above, we might be able to identify a number of genres, but it would be difficult to say how they were being adapted to and used in OD work.) With the activities of OD now defined, we can follow Spinuzzi's (2003) call to integrate multiple levels of analysis in

researching genres; in his terms, we will move from the analysis of macroscopic activity to the analysis of mesoscopic goal-directed action.

With that being said, the exact networks of genres used in OD varied from one game project to another: for example, Unknown Worlds Entertainment used an in-game player feedback tool for their game while Amplitude Studios used a voting system to allow players to select feature for their games. And the genres themselves were often subdivided down into several subgenres that were implemented in various ways: forums on both digital distribution sites and on developer sites incorporated numerous subgenres, including rules for interaction (which appeared on less than half of forums and which usually took the form of a list), player complaints (which used narrative to describe problems), and technical support questions (which redirected the forum away from OD feedback activity). In the sections below, I will describe the major genres being used in OD projects and classify them according to the activities described above. Furthermore, I will define each genre as obligatory, standard, or optional to each activity. I defined obligatory genres as the primary means by which each activity was mediated. I defined standard genres as those that were present in most OD projects, but which may not have been as central to the activities as the obligatory genres. Finally, I defined optional genres as those that less commonly used to support activities. The fact that these genres were labeled as “optional” should not be taken to mean that they were not incredibly influential in some OD projects; instead, it is meant to signify only that they were not present and influential in many OD activities. Finally, it must be noted that these categorizations are also open to movement and further formalization: I decided on each category in an inductive manner by exploring a range of OD projects, but I did not use a coding system that could have offered further validity to this study. The genres²⁵ discussed below include: chats,

²⁵ The term genre here is somewhat problematic. Some items I refer to might alternately be conceptualized as something other than a genre. For example, update announcements might be better classified as a metagenre (as a

comments, dev sites, digital distribution store pages, fan art, forums, game design documents, game iterations, in-game feedback, player-built content, production trackers, reviews (and previews), social media posts, update announcements, videos, and votes. In some cases, I will need to divide these into further subgenres in order to accurately represent the use of the texts. I will primarily be drawing examples from one OD project, *Subnautica*, because it is still in development, because the developers have completed multiple successful OD projects, and because the network of genres surrounding this games is complex and active. At the end of this section, I will offer a summary of the classifications of the genres.

CHATS

OD chats were discussions, sometimes synchronous (live) and sometimes asynchronous that were hosted on a range of platforms including IRC. In many ways, chats functioned in similar ways to OD forums, but they typically involved less structure and were less persistent across time. In some cases, developers hosted chats to gather feedback from players and to describe their work. For example, the *Subnautica* dev team hosted an ongoing chat using Discord (a voice and text chat system marketed to gaming communities). In other cases, players hosted their own chats to work on collective projects (like wikis). Typically, chats were characterized by a quick, give-and-take form of written discussion: individual comments were rarely longer than a few lines and many different people participated actively at once. Given the discursive nature of the chats, we can classify it as part of both OD design and feedback activities. On the other hand, chats were not often used for marketing questions, and consumers did not typically participate. And while they were relatively common in OD projects, they were not necessarily the most widely acknowledged

broad pattern that can be recognized across many individual genres (Carter, 2007)). On the other hand, social media and videos might be better described as media or communication channels that contain multiple genres. I will unpack some of this in the discussion of individual items below.

or promoted forms of feedback or design. As such, I defined them a standard genre used in feedback activity.

COMMENTS

Comments are probably the most ubiquitous form of online communication to the point where it is often more notable if a site blocks comments than if it allows them. OD comments were attached to several other OD genres including reviews, social media, and blogs. While comments shared some structural similarities to other player-generated genres (such as brevity), they were far more responsive in nature (i.e., they were framed as replies to existing conversations rather than initiating discourse on a specific topic). Comments served a number of functions in OD projects. In some cases, they offered compliments or complaints about the game as a whole: for example, some players commented on a post on *Subnautica's* Facebook page to say that the dev team was doing a great job and that this was among their favorite games. In other cases, comments offered specific suggestions related to the post. For example, in response to *Subnautica's* recent announcement of support for Oculus on their Steam page, several users used the comments to request support for Vive as well (Jeremy, 2016). Other comments worked in a more discursive fashion by engaging with the topics of other comments (thus mirroring the discussions on forums) by using an @username construction. Overall, the vast majority of comments were posted by players, while only a few were posted by the dev team. Many of the comments were not directly related to feedback activity, but the sheer number of such comments means that player comments can be categorized as a standard genre of feedback activity.²⁶

DEV SITES

Most OD games had a website (or a dedicated section of a larger developer website). These

²⁶ They can't be defined as an obligatory genre of feedback primarily because there is little evidence that developers regularly read these comments (as opposed to stronger evidence of developer interaction on forums).

sites served a range of purposes, but they often acted as the main presence for an OD project outside of digital distribution sites and player-built sites. As with any website, the dev sites were host to a range of subgenres that serve different purposes. Two major subgenres appeared on most dev sites: the game homepage and a dev blog. Individual dev sites were also host to a range of other genres. For example, many dev blogs included the original version of update announcements; however, these announcements were also hosted in different forms in several additional locations, making them not a true subgenre of the dev site. In the next two paragraphs, I will describe the dev blog and homepage separately since they primarily served different OD activities.

Dev blogs often acted as the primary place in which developers laid out their plans for the game and continued to update players about the progress being made toward design goals. Individual posts ranged across genres that I discuss elsewhere (including update announcements and game design documents) as well as unique content such as social updates. For example, the *Subnautica* blog post from November 2015 explained a slowdown in production due to a planned team retreat (Jeremy, 2015). It included pictures of the team and worked primarily to build a collegial picture of the group rather than to give specific information about the game. Overall, dev blogs could be described as an obligatory genre used in OD design activity.

Website home pages are a varied set of texts that have been subject to a number of genre analyses (see, for example, (Askehave & Nielsen, 2005; Eissen & Stein, 2004). Likewise, dev sites for OD projects also used home pages for a range of purposes. Some sites included spaces for player interaction (such as comments) and some included design communications (such as dev blog posts). However, home pages universally supported marketing activity. For example, the page for a new, less known game, *Shakedown Racing*, was simply a large colorful image with a link to the Steam store page. And while the *Subnautica* home page remediated recent dev blog posts and

a copy of the dev twitter feed, it also featured prominent links to the digital distribution storefront for the game. As such, we can classify home pages as an obligatory genre of marketing activity, since any other content seemed to be in support of that goal.

DIGITAL DISTRIBUTION STORE PAGES

Digital distribution sites can be broadly defined as any website where individual sellers can directly reach consumers. In the games industry, there are numerous digital distribution sites, of which Steam is by far the most popular. Several such sites included specific programs for OD projects, including Steam Early Access, GOG Games in Development, Desura Alphafunding, and Itch.io (a platform typically used by many very small indies). OD games each had a main store page on these sites, but the content varied somewhat between digital distribution sites. All such pages had a way to purchase and download the game (typically in the form of a buy button and a shopping cart), a textual description of the game (for example, *Subnautica*'s description begins with "Descent into the depths of an alien underwater world filled with wonder and peril" ("Subnautica," n.d.)), and images or videos of the game. The images included both screenshots of the game and images meant to invoke the feeling of the game (in much the same way as covers of game boxes used to do). Videos included gameplay videos and game trailers, which only sometimes used actual gameplay (these will be discussed in more detail in the section on videos below). Digital distribution store pages also often included other genres, including update announcements and reviews. These pages were universally controlled by the developers of the game (though some did include spaces for player feedback such as reviews or comments). Overall, digital distribution pages were an obligatory genre of marketing activity.²⁷

²⁷ The categorization as obligatory might be argued against by citing famous OD projects such as *Minecraft* where the dev team sold their product exclusively on their own website. These cases, however, are a rare exception in the current world of OD indie developers.

On the other hand, some pieces of the digital distribution pages also appeared to serve design activity. For example, all store pages for Early Access games on Steam opened with a text box titled “Why Early Access” where developers explained why they chose to release their game using the Early Access program. While this information, in and of itself, met the definition above of design activity, its placement served marketing more directly—current players were less likely to reference this material, and the specific content of the text boxes often worked to reassure consumers of the safety of the purchase rather than giving in-depth information about specific design choices.

FAN ART

Broadly speaking, fan art can be understood to include both 3D rendered objects (without additional functionality that would make them mods) and hand drawn art (as well as fan fiction and other forms of nonvisual art). OD fan art was similar to that of any other fandom community; however, it had the potential to act as a form of feedback from players to developers, especially in cases where the art was meant to suggest changes in the game art style. OD fan art was posted to a range of sites; for example, fan art of *Subnautica* was on the Steam community section, deviantart.com, social media and player blogs, and the wiki. However, since fan art was only rarely integrated into OD projects directly, it was an optional genre of feedback activity.

FORUMS

Forums were probably the most heavily used form of communication in OD projects. Often a single game had multiple forums. For example, major forums for *Subnautica* existed on the Unknown Worlds Entertainment website, Steam, and Reddit.²⁸ Occasionally, forums for OD

²⁸ Though it is certainly open to debate, I would argue that reddit functions primarily as a large collection of forums. In this case, *r/Subnautica* shares many functions with the other forums mentioned, though with less high level structure.

games also appeared on other sites, such as wikis, but these forums typically supported activities other than the three OD activities defined above (for example, the forum on the *Subnautica* wiki organized the player support work for the wiki itself).

OD forums were not significantly different from any other forum: they grouped discussions under general sections (for example, *Subnautica*'s forum on Steam included sections for General Gameplay Discussion; Bugs, Crashes, & Other Horror; Story & Lore; Oculus VR; Steam Economy Items; and Trading). Discussions then occurred in threads with an original poster and subsequent responders, with both players and developers acting in both roles at different points in the discussions.

Forums were used in both game design and feedback activities could be described as the place where these two activities intersected most heavily. Original posts in forums generally belonged to a limited range of subgenres: compliments/complaints, suggestions²⁹, updates, rules, support, and value queries. Compliments/complaints were broad statements about the game or development process and didn't focus on specific ideas for change. For example, complaint posts about unpopular games might claim that the game was garbage or that it was never going to be finished. These posts were alternatively directed at either developers or at other players. Suggestion posts were more focused and covered a range of feedback topic including new content or features and bug fixes. While these posts were ostensibly directed to the audience of the dev team, they were only occasionally responded to by developers. Update forum posts were similar to the update announcements posted elsewhere but were often more focused on a specific topic (such as a delay in the development process). These posts implicitly opened discussions around design decisions. Rules posts were typically a sticky or pinned topic at the top of a forum that

²⁹ We can see here a connection between the purposes of some original posts in forums and feedback comments. I will return to this concept of metagenres of feedback below.

established certain standards for communication (for example, see the discussion of “constructive feedback” and “civil discourse” as rules for feedback activity above). These were most often created by the dev team and were sometimes based on existing sets of rules (such as Steam’s rules for community interaction). Support posts offered or requested gameplay or technical support. Responses most often came from the player community, though the dev team also occasionally got involved in particularly intractable or common issues. Finally, value queries are one of the largest outliers of any OD genre since they are written by consumers. These posts occur on every major OD forum. The original post is invariably a question like “Is the game worth buying?” or “Is it worth it now?” Responses come from players who either recommend a purchase or to wait for the full release.

Given the range of topics covered and the fact that a single thread sometimes included posts belonging to several of the subgenres listed above, forums had to be classified as a whole rather than as individual subgenres. And since ubiquitous forum subgenres existed for each of the OD activities, it had to be classified as an obligatory genre for each of the three activities.³⁰

GAME DESIGN DOCUMENTS

Game design documents were typically posted by developers early in the OD process. While in some cases, developers chose to share their internal game design documents, this publically posted genre was more often created for the explicit purpose of sharing a vision of the game with the player community. For example, Jeremy (2013) introduced the concept of *Subnautica* and outlined the dev teams’ goals for the final product, including “[eliciting] a feeling of the unknown” and “[allowing players] to design, construct and crew submarine vessels.” Amplitude studios, the dev team *Dungeon of the Endless*, also developed game design documents,

³⁰ Of the three activities, forums were used least in support of marketing, but the prevalence of value queries necessitates the classification.

but they posted them in the official forums to elicit player feedback (for example, see Mysterarts, 2013a). These posts covered the background story for the game, an overview of the gameplay, and specific details about several of the planned mechanics and feature. Regardless of their location, the game design documents acted a standard genre of design—while these documents had the potential to be significantly influential, they were not ubiquitous across all OD projects.

GAME ITERATIONS

In the initial conception of OD activity above, I designated the game iteration as the object of the activity, but it also had the potential to act as a form of design communication within OD projects. For example, Luck and Day (2015) said that they used the game itself to frame aspects of player feedback:

The cycle of release, playtest, release, does a strange thing to a design. It gives you the luxury of knowing. You can eventually know if your UI is clear, or if your game is balanced, or if your progression is engaging. You just have to keep releasing the game and asking questions. You can ask players about something specific, sure, but you can also let the gameplay ask questions for you.

In this way, the game iterations were a fundamental aspect of how communication within OD projects was directed and controlled. While there is only limited evidence that developers (as a whole) explicitly viewed game iterations as design communication (rather than as something other than communication), the ubiquity of iterative releases in OD projects necessitates the classification as an obligatory genre.

IN-GAME FEEDBACK

Some OD projects built a feedback system directly into the game. It wasn't uncommon for analytics feedback to be produced automatically during play (so developers could see crashes and other major breaks), but some games also expanded beyond this to offer players the opportunity to submit feedback mid-game. For example, *Subnautica* featured a system where, at any time

during gameplay, a player could press F8 and causing the game to pause and bring up a feedback overlay. This feedback tool had spaces for textual feedback, for categorization (as general, gameplay, bug, or framerate), and for the player's emotion (as happy, okay, unhappy, or angry). *Victor Vran* had a similar in-game feedback tool that was carried over from the OD version to the official release of the game (Blizzard, 2015). While these tools had the potential to significantly increase player feedback, they were relatively rare in OD projects, making them only an optional genre.

PLAYER-BUILT CONTENT

Though it is rare, some high profile OD projects have actually integrated player-built content into the official release of the game. For example, Garriott (2014) referenced how his team encouraged players to develop background story that was then included in in-game books. Likewise Brown (2016) discussed how the *Unreal Tournament* dev team released the Unreal Editor so players could create content as well. In these cases, there was a somewhat strange shift in the activity systems where the players became directly involved in design activity while also providing a kind of deep feedback that could influence dev team priorities, making the genre a part of both design and feedback activities. Yet, it was still such a rare practice to directly use the content that it can only be labeled as an optional genre.

PRODUCTION TRACKERS

Some OD projects had specific tools that helped players observe the progress of game development. *Subnautica* actually offered two such tools. The first was a Trello board that showed the major production goals; these goals were subdivided according to whether they were future work, currently in progress, or part of past updates. The second tool was a list of changes and check-in which that tracked each team member's daily work. This list was much more detailed

and helped players see exactly what the team was working on. Other OD projects have also included production trackers. For example, Amplitude Studios used a roadmap to show the progress on each major component of a game (including art, design, code, and tuning)—this roadmap was displayed as progress bars that identified the most recent work on each component as well as the next goal. Amplitude also used a production tracker to show how they were responding to player feedback. In a forum post, Amplitude listed all player feedback and then used a color coded legend and a status section to show how and when they would be implementing the suggestion (Mysterarts, 2013b). One of the more interesting aspects of this list was how the dev team explained their design decisions, particularly when they labeled some suggestions as not compatible with their vision for the game. For example, one player recommended that they add “different upgrades by modules” to *Dungeon of the Endless*; the dev team marked this suggestion as not compatible with their vision because “[they wanted] to keep it simple” (Mysterarts, 2013b). In the cases where they have been implemented, these production trackers were an incredibly useful way for the developers to support design activity; however, given its relative rarity, it is only an optional genre within OD projects.

REVIEWS (AND PREVIEWS)

Across all websites, reviews and previews typically serve a marketing purpose: they evaluate a product based on specific criteria and make a purchase recommendation to consumers. However, as I discussed in the previous chapter, the purpose of player game reviews has begun to drift. OD reviews have drifted even further in that they often speak in anticipatory language about the final product (and thus reflect the genre of video game previews even while offering a purchase recommendation or summative rating). For example, thylsoft (2015) wrote of *Subnautica*, “This game has the potential of becoming a real milestone in game history if it is treated right. Just

imagine all the possibilities... Thumbs up.” For the OD activities as described above, the most important form of genre drift was that these reviews were sometimes written to the developer. While it was not a universal practice, some developers supported this drift by directly commenting on reviews. For example, a negative review of *Origin of Destiny* opened with “Chapter 1 review only, some suggestions:” and went on to itemize several areas of feedback, including typos in the dialogue, audio issues, controls, and gameplay (MDCT, 2016). The developer of the game responded by thanking the reviewer for feedback and then discussed specific actions he was taking to address the reviewer’s concerns (Howard, 2016). While this drift marked some OD player reviews as part of feedback activity, most reviews did not include this variation, suggesting that the genre was only standard and not obligatory to the feedback activity. At the same time, since these reviews typically did still serve their purchase recommendation function, they also acted as standard genre of the OD marketing activity.

SOCIAL MEDIA POSTS

Social media encompasses a diverse set of technologies, so posts can’t necessarily be defined as a single genre (for example, we don’t use the same genre constraints in crafting a Facebook post as we do a 140-character tweet). In other words, social media is grouped here more for simplicity sake than for its definition as a genre. Even as such, we need to explore a few subgenres as different types of posts on social media serve different OD activities.

Developers have used a range of social media to reach their player and consumer bases. For example, *Subnautica* had official accounts on Facebook, Twitter, Instagram, Tumblr, and YouTube. Most of the content pushed to these sites were remediations of content posted elsewhere—in other words, social media was primarily being used to increase the reach of information posted elsewhere. For example, the *Subnautica* Twitter account recently featured an

announcement of the new gameplay video that had been posted to Steam (Subnautica, 2016). That same video appeared on the dev team blog, the Store page of Steam, YouTube, and the Facebook page. Comments on these posts, as discussed above, then became a place of discussions between the dev team and players (many of *Subnautica's* Facebook posts have over 150 comments attached). Given that they were amplifying both design and marketing activity posts, developer social media can be seen as a standard genre for both activities.

The use of social media becomes a bit more complex when we consider player posts. As might be expected, these served an even wider range of purposes. Players posts included fan art, reviews, complaints and compliments, promotions of their own video series (or other work), etc. Certainly some of this fulfilled our definition of feedback activity, but the majority of player posts did not (they were directed at other players rather than the developer or offered no substantive comment on the game). Thus despite their quantity, player social media posts were only an optional genre of feedback activity.

UPDATE ANNOUNCEMENTS

Update announcements are any form of communication that primarily offers information on how a technology has changed in a recent update. These type of announcements have become increasingly common as many developers continue to support and update software after its initial release. In OD projects, dev teams have been very interested in discovering effective ways to craft update announcements. For example, Chris Avellone discussed the difficulty of creating update announcements in a GDC panel:

Also one other issue we had was that it was a little difficult to get a good consolidated area to provide a list of changes to the player. Like because we discovered that not everyone would read a sticky forum post, for example, listing all the changes. And even then you had to do some semantic wizardry to communicate what those changes were, without flooding the player with too much information. Actually up until this date, we're still looking for new ways to better communicate the changes to the player, and we still haven't

solved that particular issue yet. (Johnson et al., 2015)

Update announcements were likely seen as so essential because they represented a means to frame the discussion of the game within the greater OD community. For example, a well-crafted announcement could potentially help a dev team head off significant and repetitive complaints about missing features. Update announcements were effectively a metagenre: they were crafted in multiple ways and were posted to a range of communication channels, but shared a common purpose and some common features. For example, *Rust*'s most recent update announcement on Steam was a bulleted list of changes (Pearson, 2016a) with a link to a the most recent post on the dev blog site, where the changes were described in more detail (Pearson, 2016b). *Subnautica* also posted update videos, which showed changes but offered little detail or specificity. Overall, update announcements were an obligatory genre of design activity. While they may have also served a secondary marketing function (more noise was generated by frequent updates, and a game was likely to seem more appealing/trustworthy if it was updated regularly), they often gave detailed information that was focused on the interests of current players.

VIDEOS

Like social media, videos are a broad category of communication that couldn't truly be labeled as a single genre. The first way we can categorize videos is by dividing developer and player created videos.

Developers created a range of videos that served both of their primary OD activities. As described above, some of these videos are best thought of as part of update announcements, but other videos focused less on illustrating specific changes and more on actively demonstrating how the game was being made. For example, Vlambeer used Twitch to live stream their development work (using Game Maker Studio) on *Nuclear Throne* (Leone, 2014; Vlambeer, n.d.). This first set

of videos were solidly within design activity, though they were used inconsistently across projects. Some developers (often with the assistance of games journalists) also created videos that catalogued their progress on the game over time. For example, Game Informer (2014) interviewed Brad Muir about *Massive Chalice's* open development process and reviewed the work they had completed over the previous years. These videos (given their polish and location) bridged marketing and design activities, but were also relatively rare. The only ubiquitous developer use of video in OD projects was a genre that supports marketing: namely the game trailer. Basically every game (both OD and traditional) produced in recent history has also had a trailer. These videos were displayed prominently on digital distribution sites and worked to highlight either the gameplay features or feel of the game. Overall, we can broadly divide developer videos among design and marketing purposes. The design videos have largely not coalesced as genres yet (there is little consistency in how they are produced across projects), but game trailers were an obligatory genre of marketing activity.

Players also created videos to support gameplay and to offer feedback. Some players created tutorials or walkthrough videos that tried to help other players learn how to play the game (or to deal with technical issues). These videos, while common, did not fit the definition of feedback activity. However, some players also created “Let’s play,³¹” review, or preview videos that showed the gameplay in general. As is the case with reviews and previews above, these could be considered an optional genre of feedback activity. Finally, players also sometimes created feedback videos. These tended to be short videos focusing on small issues or bugs where the video medium can help to illustrate the problem (for example, see Mr. PyrOx, 2015). This subgenre wasn’t terribly common, but it could also be considered to be a part of feedback activity.

³¹ Let’s play videos are an emerging genre that try to avoid offering specific recommendations for purchase. Instead, they just show the gameplay in order to help other people see what the game is like.

VOTES

Some OD projects lowered the requirements for participation by creating polls where any player could vote for changes they wanted to see. Amplitude Studios made extensive use of this system through their Games2Gether initiative, where they allowed players to vote for different characters and content for the games. This was a relatively common genre of feedback activity, though one that was heavily influenced by designers (who created the polls and thus set range of options to be voted on), which limited the depth of player participation in ways that other feedback genres did not. Overall, voting systems were a standard genre of feedback activity.

OD GENRES SUMMARY

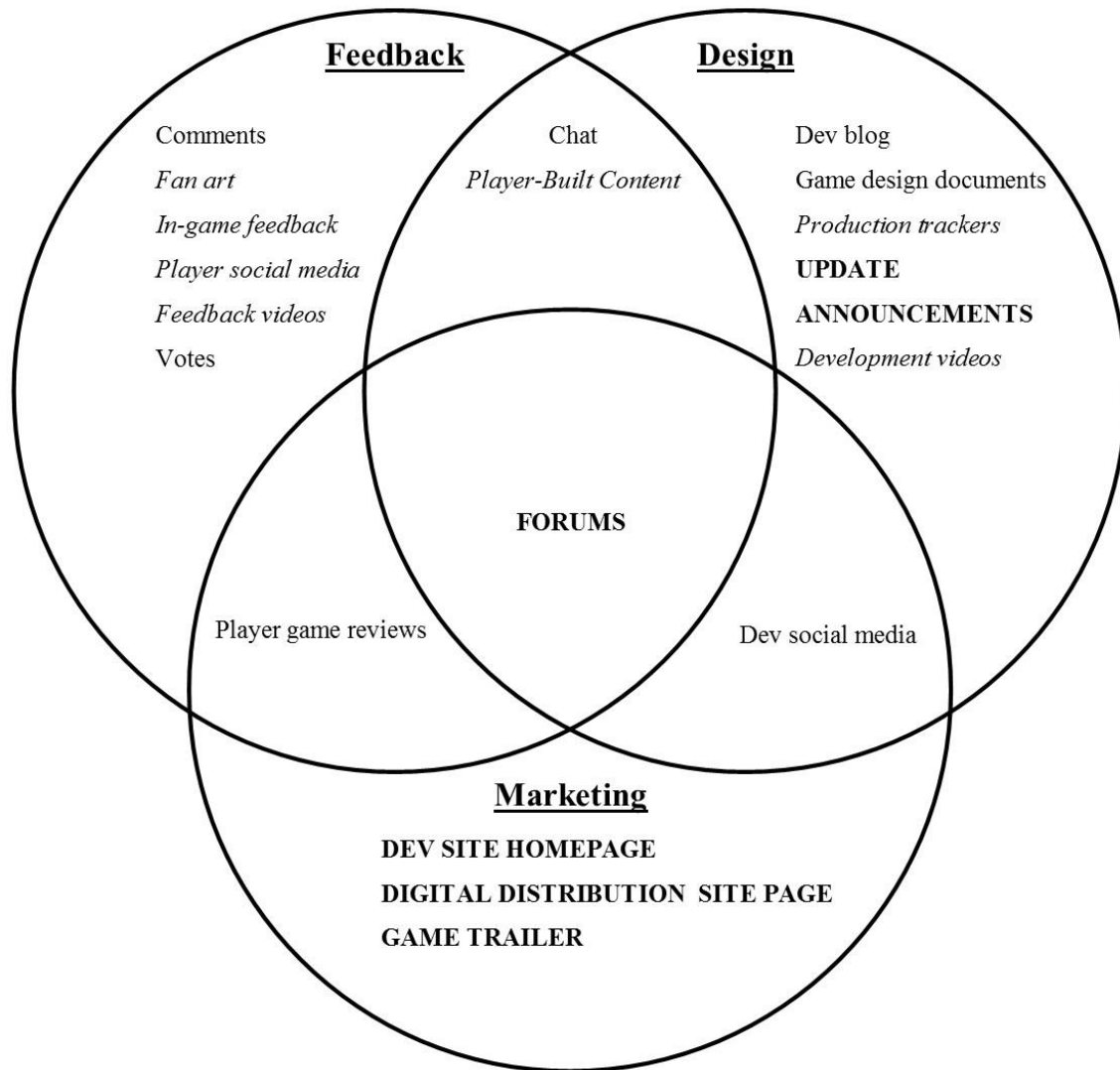
Written genres are the primary means by which OD activity is implemented. It is only through them that game development can truly be labeled as transparent or as participatory. Each OD activity was supported by a network of genres that served a range of purposes. Some of these genres acted centrally to these activities (such as digital distribution store pages to marketing), while others were only ancillary to the activity (such as reviews to feedback). Some genres initiated discourse (such as update announcements), while others remediated existing content (such as social media), and yet others only repeated or responded to discourse occurring elsewhere (such as comments). Overall, forums were probably the most central genre in connecting two of the major OD activities, but forum use was also complicated by the fact that they were spread across multiple sites (such as a game's home site and Steam).

An overall depiction of the genre network of OD can be found in Figure 8 below. I've formatted the genres names to identify them as obligatory, standard, or optional. Obligatory genres are in CAPS and bolded. Standard genres are standard black text. Optional genres are italicized.

A few patterns appeared on this figure. First, feedback activity was mediated through the

largest plurality of genres. This was likely related to decentralized production of those genres and the pattern of drift and proliferation of user-generated genres that has been noted throughout this dissertation. Second, feedback activity lacked a major genre that it did not share with design activity. This exposed one of the more interesting aspects of feedback activity: in order for players to provide extensive feedback, they had to see involvement/responsiveness from the developer, which heightened the levels of feedback on genres that the dev team also used. Finally, we can also note that forums acted as the central genre of OD activity in general: while forum posts may not have been the most important genre for OD design or marketing activities, they were the only genre to cross over all three activities.

Figure 3.8

Genres of O.D. Activities**FEEDBACK ACTIVITY AND GENRE KNOWLEDGE**

In closing, I want to return to one set of the OD activity breakdowns discussed earlier in order to explore the role of genre definition and knowledge in how we understand and address this breakdown.

While OD developers have typically painted the OD process in a positive light, they were still very aware of the potential for player feedback to overwhelm development:

It may not be wise, but we do every single thing. So we do twitter and facebook and reddit and a wiki and the forums on steam and forums on our site, and answer every single one of them. I guess that could be part of the reason why I said we were kind of overwhelmed with all the feedback and stuff, but if you direct them to the one [communication channel], it might be the same amount anyway. And it's kind of neat to be able to meet the people where they live. (Johnson et al., 2015)

The contradiction for this developer was between the desire to engage broadly with players in feedback activity and the ability to create manageable boundaries so that responding to feedback did not become the sole activity of OD. At the same time, one might also wonder about the full implications of engaging with players in these spaces of home discourse: yes, it likely created more natural and extensive communications, but it might not have been functionally more effective than a more constrained approach.

Likewise, players sometimes expressed annoyance at the feeling of being ignored by developers. For example, in an early exchange on the “Give feedback to developers” thread on the *Subnautica* Steam forums, one player complained that the developers were ignoring him. One of the developers quickly apologized and assured the player that they were not ignoring him but were “just totally swamped right now :)” (Jeremy, 2014). This kind of responsiveness had the potential to repair the damage done to player-developer relationships that sometimes resulted from breakdowns in the many-to-few feedback communications. However, these issues weren't always fixed immediately, and the resulting publicity seriously damaged several OD game projects.

There were many potential solutions to this kind of breakdown in communication, including several which developers have already tried implementing. For example, another developer in the Johnson et al. (2015) GDC panel described how his team limited their communication channels: “We try to push [feedback] toward the forums. ... Forums are really where we are trying to get people to go so that we can consolidate them.” Other developers have created tools that directed and constrained feedback (such as Amplitude Studio's Games2Gether

voting system or *Subnautica*'s in-game feedback tool). These types of responses represented work by developers to actively construct environments for feedback activity that made the object of that activity more useful and usable toward the goals of the project as a whole.

But there is also another approach that has been explored less often: namely, instructing players more explicitly on their role in the OD project and constructing realistic expectations for their work so it can integrate more smoothly into the overall activity. Perhaps the best extant example of this approach were the “rules” sticky posts on OD forums (discussed and cited above in the OD feedback activity section). In creating rules for the forums, these dev teams were effectively making genre expectations explicit. In other words (and without consciously conceptualizing it as such), they were acting in much the same way as (technical) writing instructors who endeavor to build learners' genre knowledge in order to help them transfer productive frameworks and strategies to new writing situations.

This path could be pursued further: rather than just creating rules, community managers could reconceptualize their work as, in part, helping players gain genre awareness and use genre knowledge when providing feedback in OD projects. If this type of system was implemented effectively, it could potentially distribute some of the work of maintaining effective communication away from developers and toward players: if players were offering feedback in on well-defined channels, using structures that were familiar to developers, and including all necessary information; and if they were reading other players' posts to avoid replication of points, interacting with and detailing issues where necessary, and helping to reinforce standards; then OD feedback activity might be better optimized to meet its objective while also acting more harmoniously with design activity. Of course, there would still be problems. In every discourse community, there will be different levels of engagement and differing levels of knowledge;

however, OD developers have already come to rely on some of their biggest fans (or VIPs or “stars”) to drive and regulate the conversation (see Johnson et al’s (2015) discussion of how fans were used to direct topics of conversation for *Don’t Starve* and Luck and Day’s (2015) discussion of a superfan’s role in publicizing *Desktop Dungeons*). And while some players might respond poorly to being “lectured” on how to provide feedback, that problem could likely be handled through improving the delivery of the message (rather than eliminating this content). In short, while adopting the role of discourse instructor wouldn’t help dev teams instantly solve all conflicts originating from feedback activity, it might produce worthwhile results that could help to reduce some of the current issues in the system.

The next question would necessarily be how this type of genre instruction could be implemented. The exact methods would have to vary from one OD project to the next, based on the current situation, target audience, and communication channels. Yet, the theoretical (and pedagogical) grounding could easily be informed by Writing Studies’ long history of research in how people learn new genres. Of particular interest would be technical communication work on genre acquisition in the workplace, which has generally emphasized the importance of context and socialization on novices’ ability to adopt new discourse practices (Anson & Forsberg, 1990; Artemeva, 2008; Beaufort, 1999; Dannels & Martin, 2008; Henze, 2004; Lingard & Haber, 2002; Parks, 2001; Smart & Brown, 2006). Also of interest might be the more traditional explicit forms of genre instruction that could tie directly to the instrumental goals of OD community managers engaging in this work. This pedagogical tradition often begins with the modeling of discourse and then moves through stages of practice and feedback (Macken-Horarik, 2002 qtd. in Bawarshi & Reiff, 2010). Again, this topic has been extensively studied and discussed in writing studies scholarship (see, for example, Flowerdew, 2000; Kennedy, 1983; Kotecha, 1991; Marshall, 1991;

K. Walker, 1999; Weinstein, 1987; Whalen, 1986). Regardless of the exact approach, writing studies and technical communication scholarship have a significant base of research to help OD community managers in the pursuit of this goal.

Overall, this chapter has ranged significantly in looking at the activities and genres of OD projects, but it has also shown how an activity theory analysis can help to develop a coherent image of OD work and how a study of genres focused through these activities can help us to consider specific action and change. Likewise, the chapter has sought to carve out space for technical communication within OD projects and to explicate the value of a collaboration between technical communication and OD more generally. Due to shared interests in communication between subject matter experts (developers) and users (players), OD seems like a particularly integral space for technical communication to engage with the games industry, especially as this development methodology continues to expand. It also seems like an ideal space for the academy to engage with advocacy and change within gaming communities as smaller (less-institutionalized) developers are most actively utilizing OD. These kind of emerging practices remain of the utmost importance for technical communication as it continues to explore ways to further its legitimacy and value.

CONCLUSION: PEDAGOGY OF USER-GENERATED GENRES

Throughout its expansion as an academic discipline, technical communication has struggled with its identity: calls to define and redefine the field have abounded, all while pedagogy has struggled to locate its precise goals (e.g., should it serve students through training or education, the academy as a service course, industry through client-based projects, etc.). Technical communication pedagogy has typically leaned toward an instrumental, genre-based pedagogy that simultaneously reflected a skills-training approach while also relying on definitions of the field based on reified, stable, and universal genres. For example, the U.S. Bureau of Labor Statistics (2015) said that technical communicators “prepare instruction manuals, how-to guides, journal articles, and other supporting documents to communicate complex and technical information more easily.” And the Society for Technical Communication (n.d.) dedicated a significant portion of their definition to enumerating the genres of technical communication: for example, “Software instructions help users be more successful on their own, improving how easily those products gain acceptance into the marketplace and reducing costs to support them.” While these descriptions of standard genres added specificity to the definitions, they also limited practice and pedagogy in ways that reinforced the service orientation of each.

Of course, these views have been challenged repeatedly for decades. Scholarship has tried to position technical communicators in a role of greater authority: for example, Slack, Miller, and Doak (1993) defined technical communicators as articulators; Johnson-Eilola (1996) argued for technical communication as symbolic-analytic work; and Henning and Bemer (2016) developed a flexible redefinition of the field with the goal of contributing to its power and legitimacy. Likewise, research has emphasized the changing roles of technical communicators within emerging, fluid organizational structures. For example, Spinuzzi (2007, 2015) has described how some

organizations are becoming characterized by ad hoc relations between specialists that change from one project to the next. And Zemliansky and Zimmerman (2013) questioned how we could reevaluate authorship of technical communication texts within the collaborative digital teams that characterize modern work environments. Meanwhile, pedagogy has questioned the instrumental approach for decades (R. Johnson, 1998; Miller, 1989; Spinuzzi, 1996) and has even begun to explore the limits of generalizability of practice. For example, Henry (2000) described a pedagogy built around the ethnographic study of localized workplace practice, and Hawk (2004) discussed a post-techne pedagogy that emphasized the importance of situated material being on learning.

My dissertation has contributed to this ongoing conversation by reconsidering the contributions of users to technical communication practice. And while I have used genre here as well, I have not done so in support of yet another new definition; instead, I have continually worked to remain close to my objects of study. The three genres (or genre networks) described here should not be thought of as definitions, but rather as illustrations of how we can begin to engage with traditional tasks in new ways. And as technical communication scholarship must so often revolve around pedagogical concerns, I close here by exploring the implications of emerging genres on our pedagogical practice.

Throughout the rest of this chapter, I will take up the major genres and concerns of the previous chapters individually in order to explore how they might impact our pedagogy. In each section, I will first sketch some content-oriented approaches that can be easily integrated into existing courses. Then I will turn to approaches that arise more directly for the changed practices of user-generated genres. Finally, I will conclude the chapter by offering summative thoughts on how user-generated technical communication, as a whole, might change the way we teach students.

CROWDSOURCED DOCUMENTATION WIKIS

From a traditional content angle, crowdsourced documentation wikis (CDWs) are probably the easiest of the three genres to integrate. In fact, several articles have already discussed an approach where students construct online instruction sets, a task very similar to content creation in CDWs (Johnson-Eilola & Selber, 2008; Van Ittersum, 2013). The main advantage of CDWs over these other approaches is that they can realign the topic of the work back toward technical documentation and user support (as opposed to more general, nontechnical instructions that might be present on other sites), while also offering a greater variety of content creation tasks, such as accessibility, technical editing and translation work. When constructing for accessibility, students could review existing heuristics and then construct guidelines for the construction of content. As editors, students could be create style guides and refine existing content. As translators, students could be asked to localize existing content for new audiences. Finally, we might ask advanced students to work on adapting wiki content for different media through roundtripping procedures, which could offer an opportunity for XML and/or DITA experience.

We could also begin to construct a community management pedagogy in some classes. The first aspect of this role might be the initial construction of effective authoring environments. In this role, students would not only create the kind of author support guides that I described in my chapter on CDWs, but they could also work on establishing the rules for contributors and creating tools to help create consistency. The second aspect of this role would be the active support and facilitation of the authoring community during the content creation phase. Here, students would be engaged in the day-to-day upkeep of the site and in monitoring and responding to community discussions. This would be a particularly valuable experience as organizations seek more effective ways to engage with their online communities.

The CDW approaches suggest that, through exploring complex projects, we can offer students the opportunity to engage in a myriad of compelling roles. Large projects such as these also open the potential for cross-course collaborations where more advanced students can gain experience in managerial roles as students in introductory courses are still given practice in the foundations of content creation.

PLAYER GAME REVIEWS

In and of itself, a product review would probably be considered to be too lightweight for most technical communication courses, but it need not be. Several scholars have already pointed toward the importance of educating students in the similarly simple texts of social media (see Jennings, Blount, & Weatherly, 2014; Kaufer, Gunawardena, Tan, & Cheek, 2011; Verzosa Hurley & Kimme Hea, 2014). Reviews could also represent a means to help students explore the integration of public genres with technical information. For example, students might be tasked with providing certain technical details in their reviews while using the language of reviews to make that information accessible to a broad consumer audience. Additionally, the player game review chapter explored genre drift. Reviews (or user-generated content more broadly) offer opportune spaces in which students can investigate the local construction of texts and fit their authoring choices to the reader. This could be a particularly interesting exercise with respect to reviews, where students could explore how writing for one site demands drastically different approaches than writing for another.

The player game review chapter also suggested that professional communicators may need additional methods for understanding the communication preferences of audiences on different sites. The variance model of genre analysis discussed in that chapter could help professionals study the online user communities in which their organizations' products and services are discussed.

Returning to pedagogy, students could learn discourse analysis methods in order to study the local construction of communication on various sites. This could be integrated into any number of different tasks that encourage students to consider how they would support users and communication practices in those environments. Or students could be asked to look at a particularly interesting case (such as Steam) and to consider how they would adjust the interface to meet the needs of that particular audience. In any case, this approach could encourage students to see themselves as interface designers or as user experience designers and to tie their knowledge of discourse to the support of users.

Overall, the game review chapter suggests a greater focus on discourse analysis in technical communication classrooms. The methods of discourse analysis can help students to return with specific findings based on evidence to support recommendations in professional settings. At the same time, it can help them to uncover some less than obvious aspects of a particular community's communication preferences and practices.

OPEN DEVELOPMENT

At its core, open development is closely related to the existing technical communication practices of user testing and participatory design, though with an increase in scale and a specific emphasis on transparency. Students engaging with open development might build off existing knowledge of user-centered design to consider the effects of a broader community of users/testers. This need not be restricted to game development alone, but could be a consideration in any digital technology that could be distributed widely without significant increases in manufacturing costs. But the analysis of open development activities also showed how the methodology functions at the intersections of multiple activities. With this perspective, students could explore strategies to pursue these multiple ends simultaneously. They might be asked how they would develop a

program that could balance marketing and user-feedback for specific technologies. In this case, they would want to consider how the tools and information from their feedback activity might feed into marketing/publicity activities. Or students might want to explore how to integrate design and marketing activities by building blogs that organically meet both ends (by explaining design decisions while promoting the product implicitly). In other words, turning students' attention to the broader organizational goals associated with testing and user engagement might help them to create content that is more responsive to the sometimes contradictory goals involved in technology development.

Finally, we could also turn from content development to the argument at the end of the open development chapter: to optimize feedback activities, professional communicators could turn to research on writing instruction. If we take this recommendation seriously, we would need to integrate lessons on writing pedagogy (or how communication is learned) into technical communication classes as well. In this case, students could be tasked with locating specific user communication practices online that are not serving organizational goals effectively. They then could offer recommendations on how to align user communication practices and goals more directly with the organizational preferences, while using research from Writing Studies pedagogical research to support their methods for intervention.

There are, of course, many important similarities between each of the emerging pedagogical approaches discussed here; namely, that all engage instruction in:

1. the creation of texts and environments that support and structure collaborative user-generated technical communication
2. the use of discourse analysis to uncover community-specific practices and preferences (and the subsequent use of experience architecture to match interface design to those

preferences)

3. the use of writing studies pedagogies to align user communications practices with organizational use patterns

Effectively, what these three approaches share is the use of writing studies research and approaches to reorient the content of technical and professional communication courses.

These approaches are very much in line with what is suggested by emerging practices in industry: that the roles adopted by professional communicators will only continue to expand (perhaps even to include instruction in writing), while connections between various disciplines (such as business writing and technical communication) also grow stronger. Ultimately, as communication practices become radically more accessible, many of the traditional barriers begin to dissolve. Put another way, we could say that as users increasingly create communication content, professionals must increasingly become facilitators, researchers, and instructors of communication.

APPENDIX A: REVIEW DESCRIPTORS

ID	Author	Website	Game Genre	Game
1	P Punkt Alex	Steam	Roguelike	One Way Heroics
2	Adam	Steam	Roguelike	One Way Heroics
3	Blood Flowers	Steam	Roguelike	One Way Heroics
4	Drakii ♥	Steam	Roguelike	One Way Heroics
5	Chairman of the Bored	Steam	Roguelike	One Way Heroics
6	ChristopherAmes	GOG	Roguelike	Desktop Dungeons
7	AJBAtreides	GOG	Roguelike	Desktop Dungeons
8	sirkel	GOG	Roguelike	Desktop Dungeons
9	Danster	GOG	Roguelike	Desktop Dungeons
10	summitus	GOG	Roguelike	Desktop Dungeons
11	Metsfan12694	Desura	Roguelike	Dungeons of Dredmor
12	Kamezero	Desura	Roguelike	Dungeons of Dredmor
13	Humblelinux	Desura	Roguelike	Dungeons of Dredmor
14	Quixoticrocket	Desura	Roguelike	Dungeons of Dredmor
15	Human_usb	Desura	Roguelike	Dungeons of Dredmor
16	AbyssionGaming	Metacritic	Roguelike	Teleglitch Die More
17	Luciasar	Metacritic	Roguelike	Teleglitch Die More
18	DJ_Kaas	Metacritic	Roguelike	Teleglitch Die More
19	Xenten	Metacritic	Roguelike	Teleglitch Die More
20	tayfuntuna	Metacritic	Roguelike	Teleglitch Die More
21	Arcturuss	GameSpot	Roguelike	Rogue Legacy
22	kazoo128	GameSpot	Roguelike	Rogue Legacy
23	estalling	GameSpot	Roguelike	Rogue Legacy
24	PinchySkree	GameSpot	Roguelike	Rogue Legacy
25	hunterbarnett	GameSpot	Roguelike	Rogue Legacy
26	meatsounds	Giant Bomb	Roguelike	Binding of Isaac
27	marlow83	Giant Bomb	Roguelike	Binding of Isaac
28	gamer_152	Giant Bomb	Roguelike	Binding of Isaac
29	deathpooky	Giant Bomb	Roguelike	Binding of Isaac
30	cyberwolf247	Giant Bomb	Roguelike	Binding of Isaac
31	Koko Sabre *Scruffy*	Steam	Economy Management	Sid Meier's Railroads
32	Palodin	Steam	Economy Management	Sid Meier's Railroads
33	Furanos	Steam	Economy Management	Sid Meier's Railroads
34	lovinglife139	Steam	Economy Management	Sid Meier's Railroads
35	trainguyrom	Steam	Economy Management	Sid Meier's Railroads

36	lbayres	GOG	Economy Management	Rollercoaster Tycoon 3
37	cathal	GOG	Economy Management	Rollercoaster Tycoon 3
38	carruthers43	GOG	Economy Management	Rollercoaster Tycoon 3
39	boldee	GOG	Economy Management	Rollercoaster Tycoon 3
40	ryanhstone	GOG	Economy Management	Rollercoaster Tycoon 3
41	Tajjboman	Desura	Economy Management	Gave Dev Tycoon
42	Apinkpanzer	Desura	Economy Management	Gave Dev Tycoon
43	Tdkpetrus	Desura	Economy Management	Gave Dev Tycoon
44	trun	Desura	Economy Management	Gave Dev Tycoon
45	wad67	Desura	Economy Management	Gave Dev Tycoon
46	Delcien	Metacritic	Economy Management	Farming Simulator 15
47	SassyTheYeti	Metacritic	Economy Management	Farming Simulator 15
48	MrBla	Metacritic	Economy Management	Farming Simulator 15
49	Mognut	Metacritic	Economy Management	Farming Simulator 15
50	Jkutz	Metacritic	Economy Management	Farming Simulator 15
51	Superdude028	GameSpot	Economy Management	Sim City 4
52	leonhartt	GameSpot	Economy Management	Sim City 4
53	Science2000luke	GameSpot	Economy Management	Sim City 4
54	kirkover	GameSpot	Economy Management	Sim City 4
55	texasfight54	GameSpot	Economy Management	Sim City 4
56	drakeash	Giant Bomb	Economy Management	Tropico 3
57	phaseshift	Giant Bomb	Economy Management	Tropico 3
58	shadowhawk101	Giant Bomb	Economy Management	Tropico 3

59	matiss	Giant Bomb	Economy Management	Tropico 3
60	vonsoot	Giant Bomb	Economy Management	Tropico 3
61	lesslucid	Steam	Platformer	Spelunky
62	Mattisticus	Steam	Platformer	Spelunky
63	The Gamer Formerly Known As ...	Steam	Platformer	Spelunky
64	LosingStreak	Steam	Platformer	Spelunky
65	veggie elder	Steam	Platformer	Spelunky
66	Orpheusftw	GOG	Platformer	Mutant Mudds
67	DCT	GOG	Platformer	Mutant Mudds
68	Jnrx91	GOG	Platformer	Mutant Mudds
69	McFly808	GOG	Platformer	Mutant Mudds
70	Katosepe	GOG	Platformer	Mutant Mudds
71	Infiniteeff	Desura	Platformer	VVVVVV
72	Theyeiowduck	Desura	Platformer	VVVVVV
73	GMH_Michael	Desura	Platformer	VVVVVV
74	Thecrazycsprofessor	Desura	Platformer	VVVVVV
75	Neonchicken	Desura	Platformer	VVVVVV
76	Howrus	Metacritic	Platformer	Cave Story+
77	wololoo	Metacritic	Platformer	Cave Story+
78	mrnice	Metacritic	Platformer	Cave Story+
79	jcdoll	Metacritic	Platformer	Cave Story+
80	asmodeos	Metacritic	Platformer	Cave Story+
81	Pierst179	GameSpot	Platformer	Braid
82	SydoggXxX	GameSpot	Platformer	Braid
83	browsersr	GameSpot	Platformer	Braid
84	Daavpuke	GameSpot	Platformer	Braid
85	logicalfrank	GameSpot	Platformer	Braid
86	wess	Giant Bomb	Platformer	Super Meat Boy
87	codynewill	Giant Bomb	Platformer	Super Meat Boy
88	ridash	Giant Bomb	Platformer	Super Meat Boy
89	another_silent_antago nist	Giant Bomb	Platformer	Super Meat Boy
90	mystyr_e	Giant Bomb	Platformer	Super Meat Boy
91	xlynx	Steam	Point and Click Adventure	Year Walk
92	Robotic Anxiety	Steam	Point and Click Adventure	Year Walk
93	MrNinjaSquirrel	Steam	Point and Click Adventure	Year Walk
94	Gul'dan Betrayer of Orcs	Steam	Point and Click Adventure	Year Walk
95	PsychoNerd	Steam	Point and Click Adventure	Year Walk

96	lucas90	GOG	Point and Click Adventure	Grim Fandango Remastered
97	ElijahBaley	GOG	Point and Click Adventure	Grim Fandango Remastered
98	Lovelyurine	GOG	Point and Click Adventure	Grim Fandango Remastered
99	PushingUpRoses	GOG	Point and Click Adventure	Grim Fandango Remastered
100	leeho730	GOG	Point and Click Adventure	Grim Fandango Remastered
101	Mdyter	Desura	Point and Click Adventure	Machanarium
102	Gonzodarik	Desura	Point and Click Adventure	Machanarium
103	Roloco	Desura	Point and Click Adventure	Machanarium
104	Zeb-Demon	Desura	Point and Click Adventure	Machanarium
105	Zcmander	Desura	Point and Click Adventure	Machanarium
106	WildingCarpet11	Metacritic	Point and Click Adventure	Broken Sword 5: The Serpents' Curse
107	playbill	Metacritic	Point and Click Adventure	Broken Sword 5: The Serpents' Curse
108	edmanger	Metacritic	Point and Click Adventure	Broken Sword 5: The Serpents' Curse
109	dazsin	Metacritic	Point and Click Adventure	Broken Sword 5: The Serpents' Curse
110	Storm1672	Metacritic	Point and Click Adventure	Broken Sword 5: The Serpents' Curse
111	SieraLeone	GameSpot	Point and Click Adventure	The Secret of Monkey Island: Special Edition
112	Samson089	GameSpot	Point and Click Adventure	The Secret of Monkey Island: Special Edition
113	Platyphyllum	GameSpot	Point and Click Adventure	The Secret of Monkey Island: Special Edition
114	scvido	GameSpot	Point and Click Adventure	The Secret of Monkey Island: Special Edition
115	BlackhawkBix	GameSpot	Point and Click Adventure	The Secret of Monkey Island: Special Edition
116	zamaeri09	Giant Bomb	Point and Click Adventure	The Walking Dead
117	gamer_152	Giant Bomb	Point and Click Adventure	The Walking Dead
118	branthog	Giant Bomb	Point and Click Adventure	The Walking Dead

119	ravenhoe	Giant Bomb	Point and Click Adventure	The Walking Dead
120	jslack	Giant Bomb	Point and Click Adventure	The Walking Dead
121	Sparrow_Hawk	Steam	Survival	Rust
122	Obey The Lawson	Steam	Survival	Rust
123	'Starscre4M	Steam	Survival	Rust
124	maniken50	Steam	Survival	Rust
125	k4rnifex	Steam	Survival	Rust
126	snortercle	GOG	Survival	Don't Starve
127	hayward	GOG	Survival	Don't Starve
128	gresimdave	GOG	Survival	Don't Starve
129	RiotousRyan	GOG	Survival	Don't Starve
130	Hyperboy1987	GOG	Survival	Don't Starve
131	Broax	Desura	Survival	Project Zomboid
132	Warlordpompey	Desura	Survival	Project Zomboid
133	Malakili	Desura	Survival	Project Zomboid
134	Banzaizy	Desura	Survival	Project Zomboid
135	Chris2933	Desura	Survival	Project Zomboid
136	Tobias2189	Metacritic	Survival	7 Days to Die
137	SlimPlay	Metacritic	Survival	7 Days to Die
138	thinkmodo	Metacritic	Survival	7 Days to Die
139	MR_APPLE	Metacritic	Survival	7 Days to Die
140	Scion1337	Metacritic	Survival	7 Days to Die
141	miracalious	GameSpot	Survival	DayZ
142	swavo13	GameSpot	Survival	DayZ
143	ironrice78	GameSpot	Survival	DayZ
144	Wingsteed	GameSpot	Survival	DayZ
145	mrgeorge07	GameSpot	Survival	DayZ
146	drzoltan	Giant Bomb	Survival	S.T.A.L.K.E.R.: Shadow of Chernobyl
147	mzurckerm	Giant Bomb	Survival	S.T.A.L.K.E.R.: Shadow of Chernobyl
148	star_wars117	Giant Bomb	Survival	S.T.A.L.K.E.R.: Shadow of Chernobyl
149	wemibelec90	Giant Bomb	Survival	S.T.A.L.K.E.R.: Shadow of Chernobyl
150	omali	Giant Bomb	Survival	S.T.A.L.K.E.R.: Shadow of Chernobyl
151	Klooger	Steam	Puzzle	Portal 2
152	Nailim	Steam	Puzzle	Portal 2
153	[NwA] Dreadnought [♣F2]	Steam	Puzzle	Portal 2
154	sCop3SCouT	Steam	Puzzle	Portal 2
155	Borko3fkovix	Steam	Puzzle	Portal 2
156	Overity	GOG	Puzzle	Papers Please

157	Anamon	GOG	Puzzle	Papers Please
158	nedobrak	GOG	Puzzle	Papers Please
159	strazos	GOG	Puzzle	Papers Please
160	EddieSchweitzer	GOG	Puzzle	Papers Please
161	Hackenstaben	Desura	Puzzle	SpaceChem
162	Zarius	Desura	Puzzle	SpaceChem
163	Swebban	Desura	Puzzle	SpaceChem
164	SecuirtyXIII	Desura	Puzzle	SpaceChem
165	Ca_putt	Desura	Puzzle	SpaceChem
166	genoforprez	Metacritic	Puzzle	World of Goo
167	hugamuga	Metacritic	Puzzle	World of Goo
168	Dma4o	Metacritic	Puzzle	World of Goo
169	Radio	Metacritic	Puzzle	World of Goo
170	Zombicide	Metacritic	Puzzle	World of Goo
171	CrysisFPS	GameSpot	Puzzle	Q.U.B.E.
172	ReGenesis_II	GameSpot	Puzzle	Q.U.B.E.
173	nutcrackr	GameSpot	Puzzle	Q.U.B.E.
174	EdgeJM	GameSpot	Puzzle	Q.U.B.E.
175	Anonymoe	GameSpot	Puzzle	Q.U.B.E.
176	ka_tet19	Giant Bomb	Puzzle	Antichamber
177	franticrain	Giant Bomb	Puzzle	Antichamber
178	feetoffthesky	Giant Bomb	Puzzle	Antichamber
179	dinoracha	Giant Bomb	Puzzle	Antichamber
180	deancleansoff	Giant Bomb	Puzzle	Antichamber

APPENDIX B: FREQUENCY OF MOVES AND STRATEGIES BY REVENUE TYPE

	Advertising	Digital Distribution	Total	χ^2
Total Moves	252	214	466	0.08
Move 1: Background Information Move	76	50	126	0.02
Place game in context	66	34	100	0.00
Establish reviewer's gaming experience	23	24	47	0.88
Place review in context	22	10	32	0.03
Move 2: Specific Aspects Move	72	57	129	0.19
Game design or play	75	50	125	0.03
Practical aspects	32	20	52	0.10
Game Plot	38	10	48	0.00
Developer or development practices	7	4	11	0.37
Move 3: Overall Quality Move	89	87	176	0.88
Statement of general or overall quality	68	66	134	0.86
Recommendation	47	37	84	0.28
Rating	1	10	11	0.01
Move 4 (Optional): Narrate Game Experience Move	2	8	10	0.06
Move 5 (Optional): Offer Advice Move	13	12	25	0.84
Gameplay tips	7	8	15	0.80
Advice/Suggestions for designers	6	4	10	0.53

APPENDIX C: FREQUENCY OF MOVES AND STRATEGIES BY SITE

	Desura GOG GameSpot Giant Bomb Metacritic Steam Total							χ^2
Total Moves	61	84	89	85	79	69	467	0.19
Move 1: Background Information Move	14	21	24	26	26	15	126	0.23
Place game in context	10	17	22	20	24	7	100	0.02
Establish reviewer's gaming experience	4	10	4	10	9	10	47	0.33
Place review in context	1	6	6	12	4	3	32	0.02
Move 2: Specific Aspects Move	16	27	27	23	22	14	129	0.22
Game design or play	16	23	27	27	21	11	125	0.09
Practical aspects	6	9	12	15	5	5	52	0.08
Game Plot	0	4	13	22	3	6	48	0.00
Developer or development practices	1	3	4	2	1	0	11	0.32
Move 3: Overall Quality Move	29	30	30	30	29	28	176	1.00
Statement of general or overall quality	24	24	23	23	22	18	134	0.95
Recommendation	11	12	17	17	13	14	84	0.81
Rating	3	3	0	0	1	4	11	0.15
Move 4 (Optional): Narrate Game Experience Move	0	0	1	0	1	8	10	0.00
Move 5 (Optional): Offer Advice Move	2	6	7	6	0	4	25	0.12
Gameplay tips	0	6	2	5	0	2	15	0.03
Advice/Suggestions for designers	2	1	5	1	0	1	10	0.10

REFERENCES

- 7.3. Pictures and screen captures. (n.d.). Retrieved November 23, 2014, from http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Documentation_Guide/sect-documentation_guide-style-pictures.html
- About Us. (n.d.). Retrieved April 11, 2015, from <http://www.desura.com/groups/desura>
- Albers, M. J. (2000). The technical editor and document databases: What the future may hold. *Technical Communication Quarterly*, 9(2), 191–206. <http://doi.org/10.1080/10572250009364693>
- alex_sawczuk. (2016, March 1). Expected User Conduct. *Steam Community*. [Forum post]. Retrieved from <http://steamcommunity.com/app/431600/discussions/0/405692758725553521/>
- Allen, P. R. (1996). User attitudes toward corporate style guides: a survey. *Technical Communication*, 43(3), 237–243.
- Amidon, S. R. (2005). Writing the Learning Organization: A Framework for Teaching and Research. *Business Communication Quarterly*, 68(4), 406–428. <http://doi.org/10.1177/1080569905281758>
- Anderson, P. V. (2014). Technical communication: a reader-centered approach.
- Anson, C. M., & Forsberg, L. L. (1990). Moving Beyond the Academic Community Transitional Stages in Professional Writing. *Written Communication*, 7(2), 200–231. <http://doi.org/10.1177/0741088390007002002>
- ARK: Survival Evolved. (n.d.). Retrieved March 22, 2016, from <http://store.steampowered.com/app/346110/>

- Artemeva, N. (2008). Toward a Unified Social Theory of Genre Learning. *Journal of Business and Technical Communication*, 22(2), 160–185. <http://doi.org/10.1177/1050651907311925>
- Article Feedback Tool. (n.d.). Retrieved January 10, 2015, from http://en.wikipedia.org/wiki/Wikipedia:Article_Feedback_Tool
- Article feedback/Version 5/Report. (n.d.). Retrieved January 10, 2015, from http://www.mediawiki.org/wiki/Article_feedback/Version_5/Report#Cons
- Askehave, I., & Nielsen, A. E. (2005). What are the Characteristics of Digital Genres? - Genre Theory from a Multi-Modal Perspective. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (p. 98a–98a). <http://doi.org/10.1109/HICSS.2005.687>
- Bailey, K. (2015, December 29). The State of Early Access: What it Means for Gaming, and How to do it Right. Retrieved April 12, 2016, from <http://www.usgamer.net/articles/the-state-of-early-access-what-it-means-for-gaming-and-how-to-do-it-right#comments>
- Barker, T. T. (2002). *Writing Software Documentation: A Task-Oriented Approach* (2nd ed.). New York: Longman.
- Bawarshi, A. S., & Reiff, M. J. (2010). *Genre: An introduction to history, theory, research, and pedagogy*. West Lafayette, IN: Parlor Press. Retrieved from http://wac.colostate.edu/books/bawarshi_reiff/
- Beaufort, A. (1999). *Writing in the real world: making the transition from school to work*. New York: Teachers College Press.
- Beller, J. (2006). *The Cinematic Mode of Production: Attention Economy and the Society of Spectacle*. Hanover, New Hampshire: Dartmouth College Press.
- Berglund, E., & Priestley, M. (2001). Open-source Documentation: In Search of User-driven, Just-

- in-time Writing. In *Proceedings of the 19th Annual International Conference on Computer Documentation* (pp. 132–141). New York, NY, USA: ACM.
<http://doi.org/10.1145/501516.501543>
- Bergquist, M., Ihlström, C., Ljungberg, J., & Åkesson, M. (2008, December 13). Combining Actor Network Theory and Genre Theory to Understand the Evolution of Digital Genres [Article]. Retrieved October 29, 2014, from <http://sprouts.aisnet.org/8-20>
- Berkenkotter, C., & Huckin, T. N. (1995). *Genre Knowledge in Disciplinary Communication: Cognition/culture/power*. Routledge.
- Bhatia, V. K. (1993). *Analysing Genre: Language Use in Professional Settings*. New York: Longman.
- Bhatia, V. K. (2002). Applied genre analysis: a multi-perspective model. *Ibérica: Revista de La Asociación Europea de Lenguas Para Fines Específicos (AELFE)*, (4), 3–19.
- Biber, D., Connor, U., & Upton, T. A. (2007). *Discourse on the Move: Using corpus analysis to describe discourse structure* (28 edition). Amsterdam ; Philadelphia: John Benjamins Publishing Company.
- BigPun. (2016, March 11). [READ BEFORE POSTING] Forum Rules and Guidelines. *Steam Community*. [Forum post]. Retrieved from <http://steamcommunity.com/app/410560/discussions/0/405694115204134573/>
- BitCake Studio. (2016, March 25). Leave us your feedback! *Steam Community*. [Forum post]. Retrieved from <http://steamcommunity.com/app/370770/discussions/0/392184522710616260/>
- blackknight115. (2015, August 4). Is Early Access A Good Thing For Players Or Developers? *Rock, Paper, Shotgun*. [Comment]. Retrieved from

- <https://www.rockpapershotgun.com/2015/07/31/steam-early-access-good-bad/>
- Blizzard. (2015, July 24). Victor Vran graduates from early access! *Steam*. Announcement. Retrieved from <http://steamcommunity.com/games/345180/announcements/detail/815500659488613362>
- Bowman, S., & Willis, C. (2003). *We Media: How audiences are shaping the future of news and information*. Reston, VA: The Media Center at the American Press Institute. Retrieved from http://www.hypergene.net/wemedia/download/we_media.pdf
- Brown, J. (2016, February 26). Unreal Tournament: Lessons in Open Development. Retrieved March 16, 2016, from <http://www.makinggames.biz/features/unreal-tournament-lessons-in-open-development,10264.html>
- Bruns, A. (2008). *Blogs, Wikipedia, Second Life, and Beyond: From Production to Prodisage* (1st New edition edition). New York: Peter Lang Publishing Inc.
- Byrne, J. (2005). Evaluating the Effect of Iconic Linkage on the Usability of Software User Guides. *Journal of Technical Writing and Communication*, 35(2), 155–178. <http://doi.org/10.2190/UUQL-XBRF-UKL6-MRGY>
- Carter, M. (2007). Ways of Knowing, Doing, and Writing in the Disciplines. *College Composition and Communication*, 58(3), 385–418.
- Castelnerac, F. (2014, December 18). Open Development from all Perspectives. Retrieved March 22, 2016, from <http://www.makinggames.biz/features/open-development-from-all-perspectives,7156.html>
- Category:Documentation/BeingEdited. (n.d.). Retrieved December 13, 2014, from <https://wiki.openoffice.org/wiki/Category:Documentation/BeingEdited>
- Category:Documentation/NeedsRework. (n.d.). Retrieved December 13, 2014, from

<https://wiki.openoffice.org/wiki/Category:Documentation/NeedsRework>

Chapter 2. Documentation Workflow. (n.d.). Retrieved November 14, 2014, from

<http://docs.fedoraproject.org/en->

[US/Fedora_Draft_Documentation/0.1/html/Documentation_Guide/chap-](http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Documentation_Guide/chap-)

[documentation_guide-workflow.html](http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Documentation_Guide/chap-documentation_guide-workflow.html)

Charlotte. (2015, April 9). No Laughing Matter: The Problem With Steam's "Funny" Review

Button. Retrieved from [http://femhype.com/2015/04/09/no-laughing-matter-the-problem-](http://femhype.com/2015/04/09/no-laughing-matter-the-problem-with-steams-funny-review-button/)

[with-steams-funny-review-button/](http://femhype.com/2015/04/09/no-laughing-matter-the-problem-with-steams-funny-review-button/)

Coghlan, T. (2015, March 13). To use or not to use: The benefits and challenges of offering early

access during game development. Retrieved March 30, 2016, from

http://www.gamasutra.com/blogs/ThomasCoghlan/20150313/238710/To_use_or_not_to_

[use_The_benefits_and_challenges_of_offering_early_access_during_game_development](http://www.gamasutra.com/blogs/ThomasCoghlan/20150313/238710/To_use_or_not_to_)

[.php](http://www.gamasutra.com/blogs/ThomasCoghlan/20150313/238710/To_use_or_not_to_)

Contributor's 101. (n.d.). Retrieved January 10, 2015, from

<https://wiki.openoffice.org/wiki/Documentation/Contribute>

Cooper, R. (2016, March 11). Steam Early Access Becoming a Hit with a Success Rate of 25%

[SegmentNext]. Retrieved from [http://segmentnext.com/2016/03/11/steam-early-access-](http://segmentnext.com/2016/03/11/steam-early-access-becoming-a-hit-with-a-success-rate-of-25/)

[becoming-a-hit-with-a-success-rate-of-25/](http://segmentnext.com/2016/03/11/steam-early-access-becoming-a-hit-with-a-success-rate-of-25/)

Crawley, D. (2014, June 5). Steam Early Access: Your rights when game development stops.

Retrieved March 30, 2016, from [http://venturebeat.com/2014/06/05/steam-early-access-](http://venturebeat.com/2014/06/05/steam-early-access-rights/)

[rights/](http://venturebeat.com/2014/06/05/steam-early-access-rights/)

Crowe, J. (2014). A Great Disturbance in Development: The Dark Side of Early Access. In *GDC*

Vault. Cologne, Germany. Retrieved from <http://www.gdcvault.com/play/1020874/A->

Great-Disturbance-in-Development

- Dannels, D. P., & Martin, K. N. (2008). Critiquing Critiques: A Genre Analysis of Feedback Across Novice to Expert Design Studios. *Journal of Business and Technical Communication*, 22(2), 135–159. <http://doi.org/10.1177/1050651907311923>
- DarkForge Games. (2015, January 15). Valve added a “funny” button to Steam’s... [Facebook Status Update]. Retrieved May 10, 2015, from <https://www.facebook.com/nekro.darkforgegames/posts/696723857106812>
- Daugherty, T., Eastin, M., & Bright, L. (2008). Exploring Consumer Motivations for Creating User-Generated Content. *Journal of Interactive Advertising*, 8(2), 16–25.
- de Jong, I. K. E., & Burgers, C. (2013). Do consumer critics write differently from professional critics? A genre analysis of online film reviews. *Discourse, Context & Media*, 2(2), 75–83. <http://doi.org/10.1016/j.dcm.2013.03.001>
- DinkelServerBase. (n.d.). Retrieved December 19, 2014, from <https://help.ubuntu.com/community/DinkelServerBase>
- Documentation Wishlist. (n.d.). Retrieved December 13, 2014, from <https://wiki.openoffice.org/wiki/Documentation/Wishlist>
- Dom. (2014, November 3). Steam Reviews Are 10/10 Best Feature Ever. Retrieved from <http://egmr.net/2014/11/steam-user-reviews/>
- Eggs, S. (2004). *Introduction to Systemic Functional Linguistics* (2nd ed.). New York: Bloomsbury Publishing. Retrieved from <http://www.bloomsbury.com/us/introduction-to-systemic-functional-linguistics-9780826457875/>
- Eissen, S. M. zu, & Stein, B. (2004). Genre Classification of Web Pages. In S. Biundo, T. Frühwirth, & G. Palm (Eds.), *KI 2004: Advances in Artificial Intelligence* (pp. 256–269).

Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-30221-6_20

Engeström, Y. (1987). *Learning by Expanding*. New York: Cambridge University Press.

Engeström, Y. (2001). Expansive Learning at Work: toward an activity theoretical reconceptualization. *Journal of Education and Work*, 14(1), 133–156.

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2), 189–200. <http://doi.org/10.1177/0165551512437638>

Eyman, D. (2008). Computer Gaming and Technical Communication: *Technical Communication*, 55(3), 242–250.

Fiset, J.-F. (2016, March 17). Don't forget to read other threads. *Steam Community*. [Forum post]. Retrieved from <http://steamcommunity.com/app/365160/discussions/0/385428458165903697/>

Flowerdew, L. (2000). Using a genre-based framework to teach organizational structure in academic writing. *ELT Journal*, 54(4), 369–378. <http://doi.org/10.1093/elt/54.4.369>

FoxeoGames. (2015). *Early Access is Bad for Gaming*. Retrieved from <https://www.youtube.com/watch?v=SYJt0oJSe1Y>

Frith, J. (2014). Forum Moderation as Technical Communication: The Social Web and Employment Opportunities for Technical Communicators. *Technical Communication*, 61(3), 173–184.

Gale, S. (1996). A Collaborative Approach to Developing Style Guides. Presented at the Conference on Human Factors in Computing Systems, Vancouver. Retrieved from <http://www.sigchi.org/chi96/>

- Game Informer. (2014). *Brutal Honesty: Inside Massive Chalice's Open Development*. Retrieved from <https://www.youtube.com/watch?v=69QcBi6xNRg>
- Garriott, R. (2014, March 31). Crowdsourcing for Shroud of the Avatar. Retrieved March 22, 2016, from <http://www.makinggames.biz/features/crowdsourcing-fuer-shroud-of-the-avatar,4926.html>
- Gentle, A. (2012). *Conversation and Community: The Social Web for Documentation* (2nd ed). Laguna Hills, CA: XML Press.
- Giles, J. (2005). Internet encyclopaedias go head to head. *Nature*, 438(7070), 900–901. <http://doi.org/10.1038/438900a>
- Goldberg, D., & Larsson, L. (2015). *Minecraft: The Unlikely Tale of Markus “Notch” Persson and the Game that Changed Everything* (2nd ed.). New York: Seven Stories Press.
- Graham, J. (2009, September 21). GDC Austin: Wolfire's PR Tips. Retrieved January 21, 2016, from <http://blog.wolfire.com/2009/09/gdc-austin-wolfires-pr-tips/>
- Graham, S. S., Kim, S.-Y., DeVasto, D. M., & Keith, W. (2015). Statistical Genre Analysis: Toward Big Data Methodologies in Technical Communication. *Technical Communication Quarterly*, 24(1), 70–104. <http://doi.org/10.1080/10572252.2015.975955>
- GrahamZ. (2015, February 28). Early Access, Good or Bad? *GameSpot*. Retrieved from <http://www.gamespot.com/forums/games-discussion-1000000/early-access-good-or-bad-31858721/>
- Grayson, N. (2014, May 15). Valve Slowly But Surely Making Steam User Reviews Useful. Retrieved from <http://www.rockpapershotgun.com/2014/05/15/steam-user-reviews-better/>
- Grayson, N. (2015, February 13). Why Peter Molyneux's Godus Is Such A Disaster. Retrieved March 30, 2016, from <http://kotaku.com/why-peter-molyneuxs-godus-is-such-a-disaster->

1685539932

- Greene, J., & Palmer, L. (2014). It's All Fun and Games Until Someone Pulls Out a Manual: Finding a Role for Technical Communicators in the Game Industry. *Computer Games and Technical Communication: Critical Methods and Applications at the Intersection*, 17–33.
- Hall, C. (2015, January 16). H1Z1's Early Access launch was kind of a disaster. Retrieved April 12, 2016, from <http://www.polygon.com/2015/1/16/7556677/h1z1s-early-access-launch-was-kind-of-a-disaster>
- Hamby, A., Daniloski, K., & Brinberg, D. (2014). How consumer reviews persuade through narratives. *Journal of Business Research*. <http://doi.org/10.1016/j.jbusres.2014.11.004>
- Hart-Davidson, W. (2012). What are the Work Patterns of Technical Communication? In J. Johnson-Eilola & S. A. Selber (Eds.), *Solving Problems in Technical Communication* (pp. 50–74). University of Chicago Press.
- Haupt, L. (1881). *A manual of engineering specifications and cont...* Philadelphia: J.M. Stoddart & Co. Retrieved from <https://archive.org/stream/manualofengineer01haup#page/n5/mode/1up>
- Hawk, B. (2004). Toward a Post-Techne-Or, Inventing Pedagogies for Professional Writing. *Technical Communication Quarterly*, 13(4), 371–392. http://doi.org/10.1207/s15427625tcq1304_2
- Help On Templates. (n.d.). Retrieved November 23, 2014, from <https://help.ubuntu.com/community/HelpOnTemplates>
- Help Style Guide. (n.d.). Retrieved November 21, 2014, from https://wiki.openoffice.org/wiki/Documentation/Dashboard/Help_Style_Guide#Headings
- Hennig-Thurau, T., Gwinner, K. P., Walsh, G., & Gremler, D. D. (2004). Electronic word-of-

- mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the Internet? *Journal of Interactive Marketing*, 18(1), 38–52.
<http://doi.org/10.1002/dir.10073>
- Henning, T., & Bemer, A. (2016). Reconsidering Power and Legitimacy in Technical Communication A Case for Enlarging the Definition of Technical Communicator. *Journal of Technical Writing and Communication*, 47281616639484.
<http://doi.org/10.1177/0047281616639484>
- Henry, J. (2000). *Writing workplace cultures: an archaeology of professional writing*. Carbondale: Southern Illinois University Press.
- Henze, B. R. (2004). Emergent Genres in Young Disciplines: The Case of Ethnological Science. *Technical Communication Quarterly*, 13(4), 393–421.
http://doi.org/10.1207/s15427625tcq1304_3
- hillkill. (2015, February 2). i have a feeling dayz standalone will never be finished. *Blockland Forums*. [Forum post]. Retrieved from
<http://forum.blockland.us/index.php?topic=274415.0>
- Hobbes. (2015, August 1). Is Early Access A Good Thing For Players Or Developers? *Rock, Paper, Shotgun*. [Comment]. Retrieved from
<https://www.rockpapershotgun.com/2015/07/31/steam-early-access-good-bad/>
- Holman Rector, L. (2008). Comparison of *Wikipedia* and other encyclopedias for accuracy, breadth, and depth in historical articles. *Reference Services Review*, 36(1), 7–22.
<http://doi.org/10.1108/00907320810851998>
- Horton, W. (1994). *Designing and Writing Online Documentation: Hypermedia for Self-Supporting Products*. Wiley.

- Howard, M. (2016, April 2). Review for Origin Of Destiny. *Steam*. [Comment]. Retrieved from <http://steamcommunity.com/id/Wystell/recommended/448570/>
- How to be a successful contributor. (n.d.). Retrieved December 13, 2014, from https://fedoraproject.org/wiki/How_to_be_a_successful_contributor
- Introducing Early Access. (n.d.). Retrieved March 22, 2016, from <http://store.steampowered.com/earlyaccessfaq/>
- Jennings, S. E., Blount, J. R., & Weatherly, M. G. (2014). Social Media—A Virtual Pandora’s Box Prevalence, Possible Legal Liabilities, and Policies. *Business and Professional Communication Quarterly*, 77(1), 96–113. <http://doi.org/10.1177/2329490613517132>
- Jeremy, H. (2013, December 15). Descend into the Depths. Retrieved from <http://unknownworlds.com/subnautica/subnautica-descend-into-the-depths/>
- Jeremy, H. (2014, December 17). Give Early Access Feedback to the Dev Team. *Steam Community*. Retrieved from <http://steamcommunity.com/app/264710/discussions/0/626329187106734181/>
- Jeremy, H. (2015, November 8). Slower Pace 9 - 13 November. Retrieved from <http://unknownworlds.com/subnautica/slower-pace-9-13-november/>
- Jeremy, H. (2016, April 5). Oculus Update Released. Retrieved April 15, 2016, from <http://steamcommunity.com/games/264710/announcements/detail/707437663804023692>
- johnd13. (2015, April 6). Thoughts on Steam Early Access? *GameSpot*. [Forum post]. Retrieved from <http://www.gamespot.com/forums/pc-mac-linux-society-1000004/thoughts-on-steam-early-access-31963620/>
- Johnson, R. (1998). Complicating technology: Interdisciplinary method, the burden of comprehension, and the ethical space of the technical communicator. *Technical*

Communication Quarterly, 7(1), 75–98.

Johnson, S., Avellone, C., Cheng, J., Goya, A., Campbell, C., & Sigman, T. (2015). A Thousand Voices: Open Game Development. In *GDC Vault*. San Francisco, CA. Retrieved from <http://www.gdcvault.com/play/1021755/A-Thousand-Voices-Open-Game>

Johnson, T. (2012, June 11). My Journey To and From Wikis: Why I Adopted Wikis, Why I Veered Away, and a New Model. Retrieved from <http://idratherbewriting.com/2012/06/11/essay-my-journey-to-and-from-wikis-why-i-adopted-wikis-why-i-veered-away-from-them-and-a-new-model-for-collaboration/>

Johnson-Eilola, J. (1996). Relocating the value of work: Technical communication in a... *Technical Communication Quarterly*, 5(3), 245–270.

Johnson-Eilola, J., & Selber, S. (2008). Commons-Based Pedagogies and the Social Turn in Technical Communication. *Currents in Electronic Literacy*, 11. Retrieved from http://currents.cwrl.utexas.edu/Spring08/Johnson-Eilola_and_Selber

Jurafsky, D., Chahuneau, V., Routledge, B. R., & Smith, N. A. (2014). Narrative framing of consumer sentiment in online restaurant reviews. *First Monday*, 19(4). <http://doi.org/10.5210/fm.v19i4.4944>

Kain, D., & Wardle, E. (2005). Building Context: Using Activity Theory to Teach About Genre in Multi-Major Professional Communication Courses. *Technical Communication Quarterly*, 14(2), 113–139.

Kanoksilapatham, B. (2005). Rhetorical structure of biochemistry research articles. *English for Specific Purposes*, 24(3), 269–292. <http://doi.org/10.1016/j.esp.2004.08.003>

Kaptelinin, V. (n.d.). Activity Theory. In M. Soegaard & R. Friis Dam (Eds.), *The Encyclopedia of Human-Computer Interaction* (2nd ed.). Retrieved from <https://www.interaction->

design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/activity-theory

- Kaufer, D., Gunawardena, A., Tan, A., & Cheek, A. (2011). Bringing Social Media to the Writing Classroom: Classroom Salon. *Journal of Business and Technical Communication*, 25(3), 299–321. <http://doi.org/10.1177/1050651911400703>
- Kennedy, G. E. (1983). Teaching Formal Proposals: A Versatile Minicourse in Technical Writing. *Journal of Technical Writing and Communication*, 13(2), 123–137. <http://doi.org/10.2190/856E-WECT-DKDD-6LC6>
- KHAndAnime. (2015, April 6). Thoughts on Steam Early Access? *GameSpot*. [Forum post]. Retrieved from <http://www.gamespot.com/forums/pc-mac-linux-society-1000004/thoughts-on-steam-early-access-31963620/>
- Kotecha, P. (1991). Problem Solving and Report Writing for Second Language Engineering Students. *Journal of Technical Writing and Communication*, 21(2), 165–73.
- Lanier, C. R. (2011). Open Source Software Peer-to-Peer Forums and Culture: A Preliminary Investigation of Global Participation in User Assistance. *Journal of Technical Writing and Communication*, 41(4), 347–366. <http://doi.org/10.2190/TW.41.4.c>
- Leone, M. (2014, March 17). Vlambeer details Nuclear Throne’s public development process. Retrieved April 20, 2016, from <http://www.polygon.com/2014/3/17/5520364/vlambeer-nuclear-throne-public-development>
- Liming, D., & Vilorio, D. (2011). Work for play: careers in video game development. *Occupational Outlook Quarterly*, 55(3), 2+.
- Linds. (2016, March 17). Suggestions - Please post any suggestions here. *Steam Community*. [Forum post]. Retrieved from

<http://steamcommunity.com/app/411960/discussions/0/392183857624465332/>

Lindskog, S. (2014, January 27). The Dangers of Open Development. Retrieved February 3, 2016, from

http://www.gamasutra.com/blogs/SeanLindskog/20140127/209417/The_Dangers_of_Open_Development.php

Lingard, L., & Haber, R. (2002). Learning Medical Talk: How Apprenticeship Complicates Current Explicit/Tacit Debates in Genre Instruction. In R. M. Coe, L. Lingard, & T. Teslenko (Eds.), *The rhetoric and ideology of genre: strategies for stability and change* (pp. 155–170). Cresskill, N.J: Hampton Press.

Lippincott, G. (2003). Moving Technical Communication into the Post-Industrial Age: Advice from 1910. *Technical Communication Quarterly*, 12(3), 321–342.
http://doi.org/10.1207/s15427625tcq1203_6

List Requested Topics. (n.d.). Retrieved December 13, 2014, from <http://www.wikihow.com/Special:ListRequestedTopics>

Livingston, C. (2014, September 5). The most baffling Steam reviews. Retrieved April 16, 2015, from <http://www.pcgamer.com/the-most-baffling-steam-reviews/>

Low, R. (Ed.). (1994). *Writing user documentation: a practical guide for those who want to be read*. Englewood Cliffs, N.J: Prentice Hall.

Luck, M., & Day, D. (2015). Open Development: Taking Desktop Dungeons from Prototype to Pre-Order to Steam. In *GDC Vault*. San Francisco, CA. Retrieved from <http://www.gdcvault.com/play/1022177/Open-Development-Taking-Desktop-Dungeons>

Mackay, P. D. (1997). Establishing a corporate style guide: A bibliographic essay. *Technical Communication*, 44(3), 244–251.

- Macken-Horarik, M. (2002). "Something to shoot for": A systemic functional approach to teaching genre in secondary school science. In A. M. Johns (Ed.), *Genre in the classroom: multiple perspectives* (pp. 17–42). Mahwah, N.J: L. Erlbaum.
- Mackiewicz, J. (2008). Reviewer Motivations, Bias, and Credibility in Online Reviews. In S. Kelsey & K. St.Amant (Eds.), *Handbook of Research on Computer Mediated Communication*: (pp. 252–266). IGI Global.
- Mackiewicz, J. (2010a). Assertions of Expertise in Online Product Reviews. *Journal of Business and Technical Communication*, 24(1), 3–28. <http://doi.org/10.1177/1050651909346929>
- Mackiewicz, J. (2010b). The Co-construction of Credibility in Online Product Reviews. *TECHNICAL COMMUNICATION QUARTERLY*, 19, 403–426. <http://doi.org/10.1080/10572252.2010.502091>
- Mackiewicz, J. (2011). Epinions Advisors as Technical Editors Using Politeness Across Levels of Edit. *Journal of Business and Technical Communication*, 25(4), 421–448. <http://doi.org/10.1177/1050651911411038>
- Mackiewicz, J. (2014). Motivating Quality The Impact of Amateur Editors' Suggestions on User-Generated Content at Epinions.com. *Journal of Business and Technical Communication*, 28(4), 419–446. <http://doi.org/10.1177/1050651914535930>
- Maddox, S. (2012). Confluence, Tech Comm, Chocolate: A Wiki as Platform Extraordinaire for Technical Communication. Laguna Hills, CA: XML Press.
- Maiberg, E. (2104, September 21). Devs react to Spacebase DF-9 release: "Early Access is not an 'alternative' development approach." Retrieved March 30, 2016, from <http://www.pcgamer.com/early-access-devs-respond-to-spacebase-df-9s-unexpected-release/>

- Makuch, E. (2016, January 4). Minecraft PC Reaches New Sales Milestone. Retrieved March 19, 2016, from <http://www.gamespot.com/articles/minecraft-pc-reaches-new-sales-milestone/1100-6433491/>
- Markel, M. (2012). *Technical Communication* (10th ed.). Boston: Bedford/St. Martin's.
- Marshall, S. (1991). A genre-based approach to the teaching of report-writing. *English for Specific Purposes*, 10(1), 3–13. [http://doi.org/10.1016/0889-4906\(91\)90012-L](http://doi.org/10.1016/0889-4906(91)90012-L)
- Mason, J. (2013). Video Games as Technical Communication Ecology. *Technical Communication Quarterly*, 22(3), 219–236. <http://doi.org/10.1080/10572252.2013.760062>
- McAllister, K. S. (2004). *Game work: language, power, and computer game culture*. Tuscaloosa, AL: University of Alabama Press, c2004.
- McKenna, R. (1991, January 1). Marketing Is Everything. Retrieved April 11, 2016, from <https://hbr.org/1991/01/marketing-is-everything>
- MDCT. (2016, April 2). Review for Origin Of Destiny. Retrieved April 3, 2016, from <http://steamcommunity.com/id/Wystell/recommended/448570/>
- Mehlenbacher, B. (2003). Documentation: not yet implemented, but coming soon! In J. A. Jacko & A. Sears (Eds.), *The Human-computer Interaction Handbook* (pp. 527–543). Hillsdale, NJ, USA: L. Erlbaum Associates Inc. Retrieved from http://www4.ncsu.edu/~brad_m/research/documentation03.pdf
- Miller, C. R. (1984). Genre as Social Action. *Quarterly Journal of Speech*, 70(2), 151–67.
- Miller, C. R. (1989). What's Practical About Technical Writing? In B. Fearing & W. K. Sparrow (Eds.), *Technical Writing: Theory and Practice* (pp. 14–24). Modern Language Association.
- Mirel, B. (1998). “Applied Constructivism” for User Documentation Alternatives to Conventional

- Task Orientation. *Journal of Business and Technical Communication*, 12(1), 7–49.
<http://doi.org/10.1177/1050651998012001002>
- Miss Nuro “Fishfox.” (2014, July 5). Comment on Review for Rust. Retrieved May 31, 2015, from
<http://steamcommunity.com/profiles/76561197970343422/recommended/252490/>
- Moeller, R. M., & Christensen, D. M. (2009). System Mapping: A Genre Field Analysis of the National Science Foundation’s Grant Proposal and Funding Process. *Technical Communication Quarterly*, 19(1), 69–89. <http://doi.org/10.1080/10572250903373098>
- Motta-Roth, D. (1995). Book Reviews and Disciplinary Discourses: Defining a Genre. *Proceedings of the TESOL 29th Annual Convention & Exposition*. Retrieved from
<http://eric.ed.gov/?id=ED404858>
- Mr. Dingleberry. (2013, August 19). STEAM... Please Stop with the EARLY ACESS Crap. Retrieved March 25, 2016, from
<http://steamcommunity.com/discussions/forum/1/864976115754301626/>
- Mr. PyrOx. (2015, August 2). [Subnautica] Reaper Leviathan Fireworks + Lifepod Bug. Retrieved April 3, 2016, from <https://www.youtube.com/watch?v=75xBWEAQScA>
- Mysterarts. (2013a, October 2). [DotE] GDD 6 - EVENTS, BLUEPRINTS & MERCHANTS. *Amplitude Studios Forums*. [Forum post]. Retrieved from <http://forums.amplitude-studios.com/showthread.php?19687-DotE-GDD-6-EVENTS-BLUEPRINTS-amp-MERCHANTS>
- Mysterarts. (2013b, December 19). [G2G] Amplitude’s List of Community Feedback. *Aplitude Games2Gether*. [Forum post]. Retrieved from <http://forums.amplitude-studios.com/showthread.php?20930-G2G-Amplitude-s-List-of-Community-Feedback>
- Nardi, B. A. (Ed.). (1996). *Context and consciousness: activity theory and human-computer*

- interaction*. Cambridge, Mass: MIT Press. Retrieved from <http://proxy.lib.wayne.edu/login?url=http://www.netLibrary.com/urlapi.asp?action=summary&v=1&bookid=49713>
- Ng, I., Badinelli, R., Polese, F., Nauta, P. D., Löbner, H., & Halliday, S. (2012). S-D logic research directions and opportunities The perspective of systems, complexity and engineering. *Marketing Theory*, 12(2), 213–217. <http://doi.org/10.1177/1470593111429519>
- Ong, W. (1986). Writing is a Technology that Restructures Thought. In G. Baumann (Ed.), *The Written Word: Literacy in Transition* (pp. 23–50). Oxford: Clarendon Press. Retrieved from http://www.bc.edu/content/dam/files/centers/boisi/pdf/f08/ong_article.pdf
- Orlikowski, W., & Yates, J. (1994). Genre Repertoire: The Structuring of Communicative Practices in Organizations. *Administrative Science Quarterly*, 39(4), 541–574. <http://doi.org/10.2307/2393771>
- Page Creation. (n.d.). Retrieved November 23, 2014, from <https://help.ubuntu.com/community/WikiGuide/PageCreation>
- Parks, S. (2001). Moving from school to the workplace: disciplinary innovation, border crossings, and the reshaping of a written genre. *Applied Linguistics*, 22(4), 405–438. <http://doi.org/10.1093/applin/22.4.405>
- Pearson, C. (2016a, March 31). Devblog 104. Retrieved from <http://steamcommunity.com/games/252490/announcements/detail/848174597852380746>
- Pearson, C. (2016b, March 31). Devblog 104. Retrieved from <http://playrust.com/devblog-104/>
- Peterson, M. (2004). Why Game Documentation is essential to a satisfying user experience. *Usability Interface: The STC Usability SIG Newsletter*, 11(2), 1–6.
- Pigg, S. (2013). Coordinating Constant Invention: Social Media’s Role in Distributed Work.

Technical Communication Quarterly, 23(2), 69–87.

<http://doi.org/10.1080/10572252.2013.796545>

Plafke, J. (2013, December 24). Kickstarter and early access games are ruining PC gaming.

Retrieved March 25, 2016, from <http://www.extremetech.com/gaming/173353-kickstarter-and-early-access-games-are-ruining-pc-gaming/2>

Powazek, D. (2006, April 4). Just a Thought: Death to User-Generated Content. Retrieved from

<http://www.powazek.com/2006/04/000576.html>

Prior, P. (2009). From Speech Genres to Mediated Multimodal Genre Systems: Bakhtin,

Voloshinov, and the Question of Writing. In C. Bazerman, A. Bonini, & D. Figueiredo (Eds.), *Genre in a Changing World* (pp. 17–34). Fort Collins, Colo. : West Lafayette, Ind: Parlor Press.

Redish, J., & Barnum, C. (2011). Overlap, Influence, Intertwining: The Interplay of UX and

Technical Communication. *Journal of Usability Studies*, 6(3), 90–101.

Revenent42. (2015, April 6). Thoughts on Steam Early Access? *GameSpot*. [Forum post].

Retrieved from <http://www.gamespot.com/forums/pc-mac-linux-society-1000004/thoughts-on-steam-early-access-31963620/>

Review DayZ. (n.d.). Retrieved May 10, 2015, from <http://www.gamespot.com/dayz/user-reviews/new>

reviews/new

Reviews that try to be funny must stop. (n.d.). Retrieved April 16, 2015, from

http://www.reddit.com/r/Steam/comments/291ny0/reviews_that_try_to_be_funny_must_stop/

Ritzer, G., & Jurgenson, N. (2010). Production, Consumption, Prosumption The nature of

capitalism in the age of the digital “prosumer.” *Journal of Consumer Culture*, 10(1), 13–

36. <http://doi.org/10.1177/1469540509354673>
- Rose, M. (2013, July 2). Dwarf Fortress in 2013. Retrieved March 19, 2016, from http://www.gamasutra.com/view/feature/195148/dwarf_fortress_in_2013.php?print=1
- Roth, S. (2014, January 23). In Defence of Open Development. Retrieved February 3, 2016, from http://gamasutra.com/blogs/SimonRoth/20140123/209214/In_Defence_of_Open_Development.php
- RPS. (2015, July 31). Is Early Access A Good Thing For Players Or Developers? Retrieved from <https://www.rockpapershotgun.com/2015/07/31/steam-early-access-good-bad/>
- Russell, D. R. (1997). Rethinking Genre in School and Society An Activity Theory Analysis. *Written Communication*, 14(4), 504–554. <http://doi.org/10.1177/0741088397014004004>
- Schryer, C. F. (1993). Records as Genre. *Written Communication*, 10(2), 200–234. <http://doi.org/10.1177/0741088393010002003>
- SeeMeScrollin. (2016, February 29). Rules and Guidelines for Steam Community Discussions. *Steam Community*. [Forum post]. Retrieved from <http://steamcommunity.com/app/397060/discussions/0/405692758722828540/>
- Selber, S. A. (2010). A Rhetoric of Electronic Instruction Sets. *Technical Communication Quarterly*, 19(2), 95–117. <http://doi.org/10.1080/10572250903559340>
- Shepherd, M., & Watters, C. (1998). The Evolution of Cybergenres. In *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences - Volume 2* (pp. 97–109). Washington, DC, USA: IEEE Computer Society. <http://doi.org/10.1109/HICSS.1998.651688>
- Six, J. (2015, August 24). Is Technical Writing Part of UX? Retrieved February 20, 2016, from <http://www.uxmatters.com/mt/archives/2015/08/is-technical-writing-part-of-ux.php>

- Skalicky, S. (2013). Was this analysis helpful? A genre analysis of the Amazon.com discourse community and its “most helpful” product reviews. *Discourse, Context & Media*, 2(2), 84–93. <http://doi.org/10.1016/j.dcm.2013.04.001>
- Slack, J. D., Miller, D. J., & Doak, J. (1993). The Technical Communicator as Author: Meaning, Power, Authority. *Journal of Business and Technical Communication*, 7(1), 12–36. <http://doi.org/10.1177/1050651993007001002>
- Smart, G., & Brown, N. (2006). Developing a “discursive gaze”: Participatory action research with student interns encountering new genres in the activity of the workplace. In N. Artemeva & A. Freedman (Eds.), *Rhetorical studies and beyond* (pp. 241–282). Winnipeg, Manitoba, Canada: Inkshed.
- Smith, R. (2015, February 23). Valve to Showcase SteamVR Hardware, Steam Machines, & More at GDC 2015. Retrieved April 14, 2015, from <http://www.anandtech.com/show/9003/valve-to-showcase-steamvr-hardware-steam-machines-more-at-gdc-2015>
- Society for Technical Communication. (n.d.). Defining Technical Communication. Retrieved April 19, 2016, from <http://www.stc.org/about-stc/the-profession-all-about-technical-communication/defining-tc>
- Society of Technical Communication. (n.d.). Defining Technical Communication. Retrieved December 12, 2013, from <http://www.stc.org/about-stc/the-profession-all-about-technical-communication/defining-tc>
- so_hai. (2015, April 6). Thoughts on Steam Early Access? *GameSpot*. [Forum post]. Retrieved from <http://www.gamespot.com/forums/pc-mac-linux-society-1000004/thoughts-on-steam-early-access-31963620/>

- Sparrow_Hawk. (2014, January 18). Review for Rust. Retrieved February 27, 2015, from <http://steamcommunity.com/profiles/76561197970343422/recommended/252490/>
- Spinuzzi, C. (1996). Pseudotransactionality, Activity Theory, and Professional Writing Instruction. *Technical Communication Quarterly*, 5(3), 295–308. http://doi.org/10.1207/s15427625tcq0503_3
- Spinuzzi, C. (2002). Modeling Genre Ecologies. In *Proceedings of the 20th Annual International Conference on Computer Documentation* (pp. 200–207). New York, NY, USA: ACM. <http://doi.org/10.1145/584955.584985>
- Spinuzzi, C. (2003). *Tracing Genres Through Organizations: A Sociocultural Approach to Information Design*. MIT Press.
- Spinuzzi, C. (2007). Technical communication in the age of distributed work. *Technical Communication Quarterly*, 16(3), 265–277. <http://doi.org/10.1080/10572250701290998>
- Spinuzzi, C. (2015). *All Edge: Inside the New Workplace Networks*. Chicago: University of Chicago Press. Retrieved from <http://www.press.uchicago.edu/ucp/books/book/chicago/A/bo19722830.html>
- Spinuzzi, C., & Zachry, M. (2000). Genre Ecologies: An Open-system Approach to Understanding and Constructing Documentation. *ACM J. Comput. Doc.*, 24(3), 169–181. <http://doi.org/10.1145/344599.344646>
- Spock, J. (2014). Bringing the Community into the Dev Team - A Look into Open Development. In *GDC Vault*. Los Angeles, CA. Retrieved from <http://www.gdcvault.com/play/1021475/Bringing-the-Community-into-the>
- Style Guide. (n.d.). Retrieved November 23, 2014, from http://wiki.blender.org/index.php/Meta:Guides/Style_Guide

- Subnautica. (2016, March 27). Here's a new Early Access gameplay video, with a crazy @sy1k1 soundtrack. What do you think? [https://www.youtube.com/watch?v=gZ8I_JiFiDE ...](https://www.youtube.com/watch?v=gZ8I_JiFiDE...)
[pic.twitter.com/X4eDRx6LZK](https://twitter.com/X4eDRx6LZK). *Twitter*. Retrieved from <https://twitter.com/Subnautica/status/714040742906241025>
- Subnautica. (n.d.). Retrieved March 22, 2016, from <http://store.steampowered.com/app/264710/>
- Sullivan, P. (2012). After the Great War: Utility, Humanities, and Tracings From a Technical Writing Class in the 1920s. *Journal of Business & Technical Communication*, 26(2), 202–228. <http://doi.org/10.1177/1050651911430626>
- Sullivan, P., & Flower, L. (1986). How do users read computer manuals? Some protocol contributions to writers' knowledge. *Convergences: Transactions in Reading and Writing*, 163–178.
- Sullivan, S. (2015, July 9). The Tragedy of Early Access Burnout. Retrieved March 29, 2016, from <http://mmos.com/editorials/early-access-burnout>
- Swales, J. (1990). *Genre analysis: English in academic and research settings*. Cambridge, England: Cambridge University Press.
- Swarts, J. (2015). Help is in the Helping: An Evaluation of Help Documentation in a Networked Age. *Technical Communication Quarterly*, 24(2), 164–187. <http://doi.org/10.1080/10572252.2015.1001298>
- Taboada, M. (2011). Stages in an online review genre. *Text & Talk*, 31(2), 247–269. <http://doi.org/http://doi.org/10.1515/text.2011.011>
- Te, Z. (2014, May 17). Dust to Dust: The History of Counter-Strike. Retrieved March 19, 2016, from <http://www.gamespot.com/articles/dust-to-dust-the-history-of-counter-strike/1100-6419676/>

- thylsoft. (2015, February 2). Review for Subnautica. Retrieved April 3, 2016, from <http://steamcommunity.com/profiles/76561198048087396/recommended/264710/>
- TinyOS Home Page. (n.d.). Retrieved October 18, 2014, from <http://www.tinyos.net/>
- Torres, R. J. (2001). A User Centered Design Based Approach to Style Guides. In J. Vanderdonck & C. Farenc (Eds.), *Tools for Working with Guidelines* (pp. 15–33). London: Springer. Retrieved from http://link.springer.com.proxy.lib.wayne.edu/chapter/10.1007/978-1-4471-0279-3_2
- Unknown Worlds. (n.d.). Darkness and Noise - Open Game Development. Retrieved November 18, 2015, from <http://unknownworlds.com/blog/recording-open-game-development/>
- Upgrade. (n.d.). Retrieved April 11, 2015, from <https://auth.giantbomb.com/upgrade/>
- U.S. Bureau of Labor Statistics. (2015, December 17). What Technical Writers Do. Retrieved April 19, 2016, from <http://www.bls.gov/ooh/media-and-communication/technical-writers.htm#tab-2>
- Valentaten, D. (2015, January 16). A Self-Aware Steam Adds “Funny” Button To User Reviews. Retrieved from <https://ggtriple.wordpress.com/2015/01/16/a-self-aware-steam-adds-funny-button-to-user-reviews/>
- van der Meij, H., & Carroll, J. M. (1998). Principles and Heuristics for Designing Minimalist Instruction. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel* (pp. 19–46). Cambridge, Mass: MIT Press.
- Van Ittersum, D. (2013). Craft and Narrative in DIY Instructions. *Technical Communication Quarterly*, 23(3), 227–246. <http://doi.org/10.1080/10572252.2013.798466>
- Verzosa Hurley, E., & Kimme Hea, A. C. (2014). The Rhetoric of Reach: Preparing Students for Technical Communication in the Age of Social Media. *Technical Communication*

Quarterly, 23(1), 55–68. <http://doi.org/10.1080/10572252.2014.850854>

Vlambeer. (n.d.). *Nuclear Throne Development Stream*. Retrieved from <https://www.twitch.tv/vlambeer>

Walker, J. (2014, January 23). The Ignorance Of Crowds: Why Open Development Is Crap. Retrieved August 4, 2015, from <http://www.rockpapershotgun.com/2014/01/23/the-ignorance-of-crowds-why-open-development-is-crap/>

Walker, K. (1999). Using genre theory to teach students engineering lab report writing: a collaborative approach. *IEEE Transactions on Professional Communication*, 42(1), 12–19. <http://doi.org/10.1109/47.749363>

Warren, T. L. (2001). Communicating Style Rules to Editors of International Standards: An Analysis of ISO TC 184/SC4 Style Documents. *Journal of Technical Writing and Communication*, 31(2), 159–173. <http://doi.org/10.2190/UD05-TM4K-NF7W-2KWX>

Washington, D. (1991). Developing a corporate style guide: pitfalls and panaceas. *Technical Communication*, 38, 553–555.

Wawro, A. (2015, August 3). Key takeaways from Epic’s open development of Unreal Tournament. Retrieved from <http://www.gdceurope.com/news/?p=280>

Weinstein, E. K. (1987). Using Technical Articles to Teach Technical Report Writing. *Technical Writing Teacher*, 14(2), 151–57.

Weiss, E. H. (1991). *How to write usable user documentation* (2nd ed). Phoenix, AZ: Oryx Press.

Whalen, T. (1986). A Training Checklist for Engineering Proposals. *Technical Writing Teacher*, 13(1), 18–22.

Wiesehan, R. (2014, January 10). I Promise Steam Early Access Isn’t Ruining Gaming. Retrieved March 28, 2016, from <http://www.makeuseof.com/tag/promise-steam-early-access-isnt->

- ruining-gaming-opinion/
 Wiki Editing Policy. (n.d.). Retrieved November 23, 2014, from https://wiki.openoffice.org/wiki/Documentation/Dashboard/Wiki_Editing_Policy#Creating_new_Pages
- Wiki Tasks guide. (n.d.). Retrieved December 13, 2014, from http://wiki.blender.org/index.php/Meta:Guides/Wiki_Tasks_guide
- WikiGuide. (n.d.). Retrieved November 20, 2014, from <https://help.ubuntu.com/community/WikiGuide>
- Wilson, G. (2001). Technical Communication and Late Capitalism Considering a Postmodern Technical Communication Pedagogy. *Journal of Business and Technical Communication*, 15(1), 72–99. <http://doi.org/10.1177/105065190101500104>
- Winter, S., & Mroz, L. (2014, September 12). Transparent Development for Might & Magic X Legacy. Retrieved March 22, 2016, from <http://www.makinggames.biz/features/transparent-development-for-might-magic-x-legacy,6214.html>
- Writer Guide. (n.d.). [Blender Wiki]. Retrieved November 22, 2014, from http://wiki.blender.org/index.php/Meta:Guides/Writer_Guide
- X-Play Gaming Update. (2008, March 6). Retrieved April 14, 2015, from <http://www.g4tv.com/thefeed/blog/post/683425/x-play-gaming-update-030608/>
- Yates, J., & Orlikowski, W. (1992). Genres of Organizational Communication: A Structural Approach to Studying Communication and Media. *Academy of Management. The Academy of Management Review*, 17(2), 299. <http://doi.org/http://dx.doi.org/10.2307/258774>
- Yates, J., & Orlikowski, W. (2002). Genre systems: Structuring interaction through

communicative norms. *The Journal of Business Communication*, 39(1), 13–35.

- Zachry, M. (1999). Constructing Usable Documentation: A Study of Communicative Practices and the Early Uses of Mainframe Computing in Industry. In *Proceedings of the 17th Annual International Conference on Computer Documentation* (pp. 22–25). New York, NY, USA: ACM. <http://doi.org/10.1145/318372.318388>
- Zachry, M., & Ferro, T. (2014). Technical Communication Unbound: Knowledge Work, Social Media, and Emergent Communicative Practices. *Technical Communication Quarterly*, 23(1), 6–21. <http://doi.org/10.1080/10572252.2014.850843>
- Zemliansky, P., & Zimmerman, T. (2013). Workplace Realities Versus Romantic Views of Authorship: Intellectual Property Issues, Globalization, and Technical Communication. In K. St. Amant & M. C. Rife (Eds.), *Legal issues in global contexts: perspectives on technical communication in an international age* (pp. 147–167). Amityville, New York: Baywood Publishing Company, Inc.

ABSTRACT**EMERGING GENRES OF ONLINE TECHNICAL COMMUNICATION**

by

LUKE THOMINET**August 2016****Advisor:** Dr. Jeff Pruchnic**Major:** Rhetoric and Composition**Degree:** Doctor of Philosophy

Emerging Genres of Online Technical Communication is a study of how the proliferation of non-professional participation has the potential to significantly change the shape of technical communication. More specifically, I use a genre analysis methodology to investigate three forms of user-generated content: crowdsourced documentation wikis, video games user reviews, and video game open development. In the first study, I analyze five crowdsourced documentation wikis and find systemic inconsistency in the workflow and content quality of the documentation. Subsequently, I argue that practitioners should use minimalist documentation theory to design more effective user-centered author support for the wikis. My second chapter uses Bhatian move-strategy analysis to investigate variation in the genre structure of a corpus of 180 video game reviews sampled from six websites. Based on the results, I argue that there are emerging genre variations that respond to both the exigencies of specific sites and also to new types of audiences. My third body chapter explores communication practices tied to the open video game development methodology where game prototypes are publicly distributed in order to support a more transparent development process. By mapping the activities and genres used to facilitate this development process, I argue past breakdowns in the system can be traced back to differing goals among

stakeholders and to a deficit in genre knowledge that lowers the usefulness of some communications. Finally, my conclusion discusses the pedagogical implications of these emerging genres. Ultimately, I argue that, as professional technical communicators become less involved with developing content and more engaged with supporting and managing communities that develop content, we need to reorient some of our instruction toward writing studies practices that can help students become effective communication researchers and facilitators.

AUTOBIOGRAPHICAL STATEMENT

Luke Thominet is an Assistant Professor of Professional and Technical Writing at Florida International University in Miami. He earned his B.A. in English and International Relations for the University of Southern California in Los Angeles and his M.F.A. in Creative Writing from Northern Michigan University in Marquette. His research interests include discourse analysis, video game culture, and technical communication. Luke has presented at SIGDOC, CCCC, GLS, SWPACA, and numerous regional conferences.