



Developer Discourse: Exploring Technical Communication Practices within Video Game Development

Rudy McDaniel & Alice Daer

To cite this article: Rudy McDaniel & Alice Daer (2016) Developer Discourse: Exploring Technical Communication Practices within Video Game Development, Technical Communication Quarterly, 25:3, 155-166, DOI: [10.1080/10572252.2016.1180430](https://doi.org/10.1080/10572252.2016.1180430)

To link to this article: <http://dx.doi.org/10.1080/10572252.2016.1180430>



Published with license by Taylor & Francis©
Rudy McDaniel and Alice Daer



Accepted author version posted online: 20
Apr 2016.
Published online: 20 Apr 2016.



Submit your article to this journal [↗](#)



Article views: 59



View related articles [↗](#)



View Crossmark data [↗](#)

Developer Discourse: Exploring Technical Communication Practices within Video Game Development

Rudy McDaniel ^a and Alice Daer^b

^aUniversity of Central Florida; ^bThomson Reuters

ABSTRACT

This study examines the discourse style of managers, developers, engineers, and artists working for an independent game development studio. Fourteen employees were interviewed, and then the results were coded and analyzed using an exploratory, single-case case study methodology. The authors argue that the texts, tactics, and technologies used by these professionals reveal insights into the practical, outcome-oriented dimensions of technical communication within the games industry as well as deeper cultural characteristics of this community.

KEYWORDS

Collaboration; digital games; digital technologies; documentation; engineering communication; help systems; knowledge management; management practices; organizational communication; workplace studies

Exploring developer discourse

The game industry is built on dirty tricks. I mean, really, when you look at a game, a game is all smoke and mirrors in so many ways.

—n-Space employee

This article studies the technical communication practices of n-Space, an independent video game development studio, and documents how its employees communicate and share knowledge. Games produced by companies such as n-Space are created in intense, pressure cooker-like environments in which complex software systems are designed and developed by large teams, under strict deadlines, using minimal resources. The four specific research questions investigated in this article are as follows:

1. What conflicts and constraints do employees experience in their roles at this company?
2. How do they articulate the contexts of those conflicts and constraints?
3. What tools, tactics, and texts do these employees use to assist in their communication and resolution of conflict?
4. What can technical communicators learn from an in-depth case study of video game developers in their working environment?

These research questions emerged from the exploratory model shown in [Figure 1](#). We were most interested in the relationship between workplace challenges and technical communication strategies. We also wanted to understand better, from an emic perspective, the ways that professional game developers perceive the contexts, constraints, and conflicts affecting their work. Finally, we sought to better understand the materials and procedures used by developers. How did particular informational contexts shape—and become shaped by—those tools, tactics, and texts?

We argue that challenges and constraints are workforce factors that dramatically shape technical discourse in game development and further that we can better comprehend the culture of a studio by

CONTACT Rudy McDaniel  rudy@ucf.edu  School of Visual Arts and Design, University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816.

Published with license by Taylor & Francis. © Rudy McDaniel and Alice Daer
This is an Open Access article. Non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly attributed, cited, and is not altered, transformed, or built upon in any way, is permitted. The moral rights of the named author(s) have been asserted.

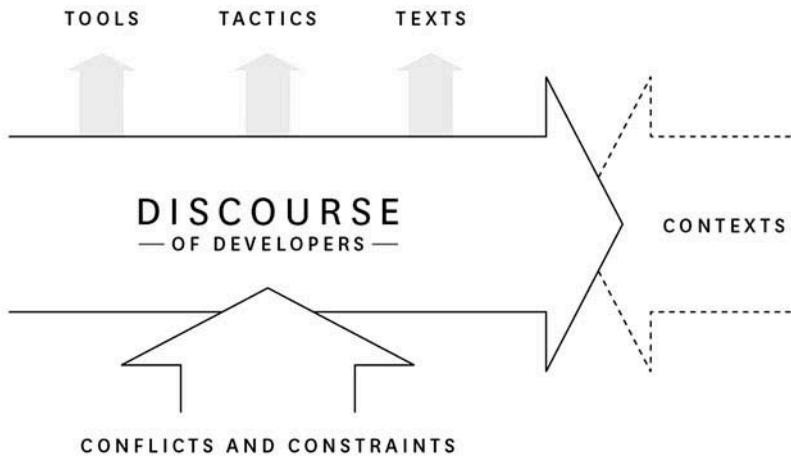


Figure 1. Exploratory research model.

studying how the studio responds to such forces. This research explains underlying mysteries of this unique community by leveraging challenge and constraint as analytic themes. It demystifies some of the smoke and mirrors used in game development, particularly in regards to the communication practices of the “communities of specialists” (Cohendet & Simon, 2007, p. 588) that craft video games. Examining the stressors surrounding the development of independent games allows us to learn from the documentation used within development as well as the communication practices that surround these texts.

Understanding these challenges and how they are addressed within this type of workplace is important because this knowledge provides insights for educators about the type of work future technical communicators may be doing in media-rich environments. Further, it suggests new research opportunities for considering how technical communication professionals might add value to improving communication within such locations. For practitioners, this better understanding can inform the design of better workflow processes and procedures for technical communicators working in environments similar to those analyzed here. Before presenting our case study results that lead to a better understanding of such matters, however, we first briefly contextualize game development within the field of technical communication.

Game development as technical communication

Developing video games requires participating in a number of challenging production scenarios, which we can consider through the lens of technical communication (Greene & Palmer, 2014). Operationally speaking, video games are complex systems, bound by rules and sustained by conflict. They are best understood holistically in combination with the playful interactions of their audiences (Salen & Zimmerman, 2004) and are filled with opportunities for exploring the relationships between diverse audiences and complex, interactive texts.

A growing body of work over the last decade considers the relationship between video games and technical communication. McAllister’s (2004) work is notable for outlining the rhetorical exigencies found in game development. Eyman’s (2008) research described the symbolic-analytic nature of game activities and the complex rhetorical situations within which players and designers solve problems. McDaniel (2009) wrote of the usefulness of games for providing procedural training, and Lamberti and Richards (2012) considered gaming communities rhetorically, arguing that gaming communities are constructed discursively through “the complex of play-oriented rhetorical activities we refer to as gaming/writing” (p. 482). Mason (2013) noted that gaming studies inform us about specific discursive strategies in communication practice, interface design, and technical

genres. Most recently, deWinter and Moeller's (2014) edited collection assembled work examining the intersection of video games and technical communication by exploring areas such as documentation, design, and privacy from various rhetorical and sociocultural perspectives. As these studies indicate, games can be considered through the lens of technical communication in many different forms and for multiple purposes.

In addition to games as artifacts of study, game development practices also raise important questions for the field. Development encompasses many specialized tasks, including world design, system design, content design, game writing, level design, and user interface (UI) design. Thus, though one's title may be "game designer," the design work that employee does varies depending on factors such as experience, aptitude, game genre, and exigencies within the development pipeline. Prior scholarship (Daer, 2010; McAllister, 2004; O'Donnell, 2009; Robison, 2008) explores aspects of development discourse, but there is still much to learn about professional game development's relationship to technical communication.

For instance, we can investigate the role of writing in game development and the multimodal design practices of hybrid teams of artists, engineers, producers, designers, and managers. Many of these employees work in the types of environments Spinuzzi (2015) characterized as "all edge adhocracies," organizations flexible and agile enough to constantly change direction, build and rebuild teams, and organize priorities around specific projects and changing digital technologies. Exploring the challenges that emerge from this dynamic informs us about how these employees communicate, how they solve problems in complex rhetorical spaces, and how they use specialized technical documents and knowledge management strategies to meet goals in fast-paced, constantly evolving workplace environments.

The best way to analyze and consider the technical communication practices of developers is to directly observe the professionals working in game studios. Due to the proprietary, competitive nature of game development, obtaining invitations to such workplaces is challenging. However, independent studios provide access to the same professional developers but often in an environment that is more welcoming for researchers. These studios are useful locations for collecting data to help us understand games as products designed, in part, by technical communication. Many of these cutting-edge technical communication practices used by video game developers in their work are broadly useful, even outside the field of game development, for working with complex data sources and interactive multimedia texts. In video game development studios, however, they are critical. We provide background about our current studio of focus, n-Space, in the next section.

Contextualizing the site: Background and major features

n-Space is an independent game development studio in Orlando, Florida. Its mission is "bringing world-class games to the masses," and its website boasts of a studio that is "better. stronger. independent" (n-Space, 2015). Employees work from offices and open floor space at a large office complex near the SeaWorld theme park located in Southwest Orlando. Seven employees founded n-Space in 1994. At the studio's height near the end of 2008, it employed 122 people.

At the time of this study, 49 employees worked at n-Space. Six employees worked in administrative roles such as human resources, accounting, or office management. The others were divided fairly evenly between engineers, designers, and artists. Of these 49 current employees, only three employees were female, and two of these individuals worked in administration. Only one of the studio's 43 production employees was a woman. The company does not track diversity information, but the employees were overwhelmingly White and male. When asked about this, the CEO noted:

On the diversity thing, all I can say is that the dramatic majority of the resumes that we get are male, you know—White male . . . Hispanic male a little bit. We've had several African Americans work here. We've had as many as five or six women in the development group. But yea, it's ridiculously lopsided in terms of White males, but not surprising considering the industry and interest.

Ethnic and gender diversity is a significant challenge in the entire game development industry, and it is worth noting up front that the communication contexts described in this article are taken almost entirely from the perspective of White males.

n-Space created more than 40 video games over the last two decades. The studio is best known for its work on titles developed for the PlayStation console, Nintendo Wii, and Nintendo handheld systems such as the DS and 3DS. The official corporate biography (n-Space, 2015) lists a number of commercial successes and recognizable releases including *Toy Story 3*, *TRON: Evolution*, *GoldenEye*, and *Call of Duty: Black Ops*. An analysis of the n-Space Twitter account reveals other games developed by the studio including *Bugriders*, *Jaws*, *Heroes of Ruin*, and *Skylanders Giants*. n-Space describes its latest title, *Sword Coast Legends*, as a *Dungeons & Dragons* video game that “brings the roleplaying dynamic between players and Dungeon Masters to life with DM Mode, a first-of-its-kind real-time experience in which Dungeon Masters guide players through unique customizable adventures” (n-Space, 2015).

During his interview, the CEO was asked about the origins of the company name. He described the importance of staking out ground around a particular technical competency and the selection of a name that supported that core institutional value. He explained:

It's a mathematical term—a generalization of any multidimensional space, one space being a single dimension, two space being x and y , three space x , y , and z When we founded n-Space, 3D graphics was the thing. That's what got us the work because we had expertise there that not many other people had. PlayStation 1 was launching. It was the first polygonal rendering hardware. So we thought the whole 3 space thing was cool, but we wanted to be n-Space to suggest the potential beyond that.

The studio name thus encapsulated the existing expertise of its founding employees, but it also spoke to a future of new possibilities and cutting-edge work. To document such cutting-edge work and the complex technical communication practices surrounding it, we used qualitative methods to investigate our research questions.

Method

We used an exploratory, single-case case study methodology. This qualitative research model is well suited for capturing the social and technical factors at work within complex communication environments (Pan & Scarbrough, 1999). In an effort to understand the ways that this group of game developers made meaning from their behaviors, we analyzed workplace documents (e.g., company organizational charts, company bios, and memos) and conducted in-depth, semi-structured interviews (Seidman, 2013). To ensure rigor, we collected a comprehensive data set with a broad participant pool of at least two different employees from five different classifications of development work. Participants were interviewed one-on-one in a conference room within their studio over a 3-month period.

Research questions were as follows:

1. What conflicts and constraints do employees experience in their roles at this company?
2. How do they articulate the contexts of those conflicts and constraints?
3. What tools, tactics, and texts do these employees use to assist in their communication and resolution of conflict?
4. What can technical communicators learn from an in-depth case study of video game developers in their working environment?

Table 1. Categories of interviewed employees at n-Space.

Category	Employees
Designers	Bob, Kevin, Sam
Engineers	Tim, Taylor, Paul, Gary
Artists	Lance, Thomas
Producers	Bryce, Beau
Managers	Carl, Joseph, Jim

Participants

Primary source materials included recorded interviews with 14 designers, artists, managers, and engineers. Employee names were anonymized for privacy purposes. Their pseudonyms, along with general job categories, are shown in [Table 1](#).

Data collection

Narrative and non-narrative survey questions (see the appendix) were administered to participants to collect responses useful for answering our research questions. Responses were digitally recorded in person on-site at the n-Space studio. The interviews were conducted by the first author, who also signed a nondisclosure agreement (NDA). The interviews were conducted with each employee individually without managers or other employees present. They were later transcribed for coding and analysis.

Each interview varied in duration from approximately 30 to 60 minutes. Many different types of employees were interviewed, ranging from the most recently hired game designer to the president of the studio. These individuals provided perspectives on technical communication from different backgrounds, including art, design, engineering, management, and production.

Data analysis and coding

We used qualitative data analysis software, QSR International's NVivo 10, for coding and analyzing the interviews. As we coded, we recognized that each useful term "assigns a summative, salient, essence-capturing, and/or evocative attribute" (Saldaña, 2013, p. 3). With this in mind, we first applied broad-brush, bucket coding to each interview to identify salient portions that connected to our research questions. This strategy allowed us to code for items such as artifacts (texts), mechanisms (tools), and strategies (tactics). After bucket coding, we constructed preliminary nodes in NVivo and built categories for organizing interview segments. These nodes organized relevant themes and clusters of conflicts/constraints across the entire set of interviews.

Guiding theory

We used Gee's (2014) Making Strange tool for discourse analysis to guide our approach to organizing and considering units of themed data. This allowed for a sophisticated qualitative analysis that included close, coded reading as well as a more distanced, outside perspective of the data. The initial categories used for preliminary grouping are shown in [Table 2](#).

Making strange: Categories of challenge and complexity

We used an iterative, recursive strategy of coding and recoding for our next level of data analysis and theme identification. We first coded for major sources of communication challenges then revisited each of these items to code for additional specificity. Challenges emerged from specific constraints, which shaped how design and communication unfolded, and/or conflicts, which introduced sources of tension, disagreement, redirection, or uncertainty into design scenarios.

Table 2. Bucket coding categories.

Node name	Description
CommStory	Provided anecdotes or brief narratives of interesting communication scenarios within the studio. Also often revealed aspects of the context from which these scenarios emerged.
Conflict	Indicated a moment or multiple moments of tension within the work environment, often requiring intervention using a tool or tactic.
Constraint	Indicated constraints which shaped how games or game components were designed and communications were influenced.
JobTitle	Used to mark the specific job titles of employees.
OrgInfo	Used to describe portions of the interviews with particular relevance to describing this organization, such as an employee's indication of how the studio is organized or who its most senior employees are.
Tactic	Identified tactics used for solving communication problems, particularly in regards to overcoming constraints.
Text	Described communication artifacts used for specialized purposes.
Tool	Described a specific communication tool used for design tasks.

Gee (2014) noted that the first step in using his Making Strange tool for discourse analysis is to consider the community as an outsider might. He wrote:

Ask yourself: What would someone (perhaps, even a Martian) find strange here (unclear, confusing, worth questioning) if that person did not share the knowledge and assumptions and make the inferences that render the communication so natural and taken-for-granted by insiders? (p. 19)

We therefore sought to unpack those situated technical communication practices that, though familiar to the designers, would be unfamiliar to outsiders.

Following this approach, we further organized conflicts and constraints into thematic categories such as “cultural,” “design-based,” “financial,” “personnel-related,” and “technical.” Each category spoke to challenges faced by game developers; oftentimes these challenges directly influenced the selections of texts, tactics, and tools used by individuals to communicate and work effectively. We recognized that some categories such as “financial” and “personnel-related” were less strange than others. These types of challenges tended to be more frequently present in other types of development environments, too, so they were less useful for helping us understand the unique discourse of game development.

After setting aside more common forms of conflict and constraint, we focused on three specialized categories of challenges that we felt revealed specific insights about the distinctive nature of game development. These included cross-disciplinary challenges, cross-cultural challenges, and technical/design challenges. We ran a query to extract the relevant interview excerpts that fit those criteria, then closely analyzed the tools, tactics, and texts from those portions of the interviews. In the next section, these three primary types of challenges are discussed and annotated with specific examples excerpted from employee interviews.

Results: Making strange the world of game development

Cross-disciplinary challenges emerge due to differences in expectations or goals between professional roles. For example, an artist desires graphical fidelity, but an engineer wants to reduce the polygon count for faster rendering. Interviews from Paul and Taylor, engineers, revealed such tensions:

Paul: I was the one teaching everybody how to do the special rigs and the other sort of effects. . . , these guys who were coming from the more traditional standpoint of art weren't able to get at this at all, so I kind of bridged the gap there.

Taylor: There's always conflicts between art and engineering. They want tools to work this way. They want to do the minimum work possible. And sometimes we can make that happen by spending a lot of engineering time developing some nice neat tool for them. . . ; it's purely resource based, so we can save the art team time by spending engineering time and vice versa.

These two excerpts reveal specialized texts such as special rigs and effects that highlight unique work done by game developers, work that requires artistic and technical expertise. Paul noted that he “was the one teaching everybody” and bridged the gap between traditional art and engineering, suggesting a perceived division between art and engineering culture. The specialized engineering tools Taylor mentions are useful for speeding up the production of art assets, but at the expense of engineering time, leading to dilemmas that must be negotiated. Word choice here characterizes time as currency that can be spent in various ways, depending on priorities, and that has a cost and an exchange value. Art time can be bought, but only by spending engineering time. Other phrasings are also revealing, such as the description of artists who want to “do the minimum work possible,” which can be facilitated if the engineer can “build a nice neat tool for them.”

Cross-cultural challenges

Commercial video game development is cross-disciplinary in practice and when games are released. Localization strategies require developers to plan for delivering games in their own country’s language and cultural context as well as for audiences in other markets. Studio employees experienced cross-cultural challenges when designing for international audiences:

Taylor: In German, a lot of the words have many more characters in them. So they require more space on screen. So there were a lot of cases where we had to expand and alter out the user interface to be able to fit all of the languages.

Kevin: On the last project, whenever we had to write words that would show up on the screen, we had to do an Excel doc called Localization, and then there is some kind of script that would read that and then decide on what country you’re in, and it would pull those messages from that Excel sheet. Like one message would be, “This is opening message number 6” and then it would be English, Spanish, Italian, whatever, and then it would take that.

Thomas: We had a character at one point who was armed with a knife, and that was a very sensitive thing because there had been an incident, a school where someone had attacked a bunch of school children with a knife, so that was something that they didn’t want to put in a game. Or there was another school situation where people had been poisoned and we had a moment where a character was putting poison in the soup and that was a no-no as well. So I think it’s interesting to balance it for different cultures.

Cultural challenges can influence the design of practically any game asset as these excerpts reveal. In Taylor’s case, localization required changes to the words and characters used within custom game components such as UIs and menus. Kevin spoke to the use of a specific Microsoft Excel document used to track localization text. With Thomas’s examples, the texts were more broadly characterized as specific scenarios and playable, in-game moments. Tools included the spreadsheet software referenced by Kevin as well as the “customized script” that was used to access that spreadsheet data and select appropriate text based on language. Taylor mentions that the UI needed to be designed more flexibly so that expansion of text was possible. Finally, tactics extracted from these samples speak to low-level, mechanical procedures for communication, such as automated text extraction from a spreadsheet, as well as more global strategies for applying cultural sensitivity to content selection and scripted interactive moments within the game levels. Thomas’s two examples of in-game events deemed acceptable in one culture but not in another reveal that cultural issues are not always universal but can be locally situated and heavily influenced by current events. Successful development strategies must be flexible at the low-level of design as Taylor’s and Kevin’s interviews illustrate, as well as at the broader level of cultural sensitivity as Thomas’s experiences indicate.

Design and technical challenges

Design challenges require developers to build game assets within specific parameters to fit certain requirements. Technical constraints require employees to modify their design plans or develop creative solutions for solving problems. Often, these challenges emerge as hardware or software limitations within certain types of video game consoles. Game development is naturally constrained by the game's design and the descriptions of the primary game mechanics, rules, and other relevant information from a game design document. However, other types of constraints present interesting technical communication challenges for the employees. For instance, conflict is introduced when required information is not readily available for some required technical production task. The excerpts from Taylor and Gary below reveal two different types of design challenges typical during game development:

Taylor: The color red was specifically reserved for a certain sect of people in the world. And so even though we didn't even have those people in the world and our game was kind of an offshoot, they were really specific. Even if it's not on a person, there can be no red. No red in the UI. No red anywhere except on these particular people.

Gary: Like with the path finding stuff, it didn't have great documentation, or really any documentation, so it was kind of difficult to—basically you had to search through the forms and through the source code and take some of the bits of information here and there and try to piece together what needed to be done to get it working.

Other excerpts reveal idiosyncratic technical challenges related to the rapid pace of changing technologies:

Paul: It came out on PC originally, yep. We were kind of limited by a couple things in that project. The hardware for one, we were limited by RAM, and we couldn't do everything that the PC could do, because let's face it: It's a 3DS this big, you know, and we were hurting with rendering what we had to render, so we had to cut down a little bit on it.

Taylor: We've also had to get to know other basic console platforms like the 360 and the Wii and the PS3 and now we're going through another console cycle where I'm sure we'll have to get on those at some point, but the big disruption is the mobile devices.

Taylor: He preferred to do things using kind of clever and inventive techniques in C++ that they got the job done and a lot of times it resulted in a lot less code, but it was extremely difficult to maintain for anyone else.

Bryce: Before I started here we weren't even using Mantis to do some type of project management work and enter in tasks and stuff like that. Before that we were using like Excel and Word docs and it was like kicking it old school back then, right? So we went from printing out tasks on paper and cutting them with a paper cutter to then moving into Mantis into the digital world.

Paul: And we had our own wiki that just had all the details of the projects in there, but yea ... it's almost a full time job for someone to keep up on maintaining the wiki, or maintaining that kind of thing, and we and we try to do it whenever we can, so.

The above excerpts reveal technical challenges that directly affect how employees develop and communicate. The first two excerpts from Paul and Taylor indicate specialized technical texts of concern to game developers, such as "what we had to render," that are directly constrained by the tools at hand, such as computational resources (RAM, or computer memory) or hardware (PC, Nintendo's 3DS). Different hardware systems have different technical limitations and developers are often tasked with the complex job of porting (translating) software from one system to another. Technical challenges also emerge from digital writing styles as Taylor's second quote explained. In this case, one employee's tactic for writing computer code favored efficiency, but counterintuitively, such tactics reduced overall organizational efficiency because no one else could understand or

maintain the code. Lastly, Bryce and Paul's final two quotes note how project management tactics have changed to accommodate larger and more complex projects. Here, texts include items such as "project management work" and "tasks and stuff like that" as well as Excel and Word docs, "tasks on paper," and "projects." Tools included specialized bug-tracking software, Mantis, community-writing spaces, such as "our own wiki," and as well as more familiar office productivity software such as Word and Excel. Particularly interesting to note is how the wiki was described as "our own wiki" rather than "the wiki," language that suggests community buy-in to this tool and the associated tactic of keeping it maintained and current.

Discussion

The three themed categories above highlight examples of technical communication texts, tactics, and tools in use within problem-solving scenarios, even if the interviewed employees might not use these terms to describe their work. However, we can also consider these elements more generally within the studio to identify other trends. For instance, texts broadly considered included forms of written documentation used by the studio as well as interactive assets, art models, programming scripts, and other documents used during the course of design and development. To identify those texts most central to the employees' professional lives, we ran a word frequency analysis on coded instances of texts to identify the words mentioned most often. As expected, games themselves constituted the most frequently mentioned texts by employees, but closer analysis of the word frequency distribution data for specialized "strange" texts also generated some useful findings. For example, one employee noted the use of "franchise guides," which functioned similarly to style guides, and indicated the appropriate contexts in which certain items from licensed properties could be used within video games. Another designer talked about specialized forms of documentation, such as the "agro system guide," which specified the rules that determined the methods by which virtual enemies attacked the player character.

Similarly, reviewing interviews for mention of specialized communication tools and tactics indicated a number of strategies used by the studio to facilitate rapid and effective communication during design and development. Knowledge of such techniques is useful for training emerging technical communicators who will find work in these types of hybrid, flexible, organizations such as the "ad hoc" discussed by Spinuzzi (2015). At n-Space, these included strategies such as the following:

- Relying upon an "open, 'flat' style of communication" for daily operations. This made it easy for anyone to move around and talk to anyone else, regardless of rank.
- Favoring maintainability and ease of use over the outright technical efficiency of workplace texts. For example, Taylor described his preference for maintainable documentation, noting, "Even if it's a little less efficient long term over a project, I think it's better to make things very clear, very easy to use and maintainable. Over all these different projects I've worked on, in the long term it ends up saving a lot of time, especially if you've got new people coming on that now have to use your system."
- Using modular design patterns that enabled texts to be reusable in future design scenarios. Taylor spoke about the importance of this tactic in his interview: "Over time, in game development, at least, you see this sort of pattern happening. In that, you'll do little bits of work here, little bits of work here and here and maybe it gets in the product and maybe it doesn't. But a lot of those end up being useful later and so you know your work lives on even if in a different place or different form that you originally intended it for."
- Implementing flexible negotiating tactics in high-stakes communication interactions to move business forward. Joseph described the importance of this rhetorical tactic when pitching new game ideas: "I got a call yesterday from somebody, from one of 'em saying yea, we've got a publisher. We've got this particular product. It needs to be done like this, in this timeline."

Here's the budget range ... gives a proposal ... and we go in and do our pitch, and sometimes they'll like what we have. Sometimes they'll want to [do] something a little bit different based on what we've got, or sometimes they'll say, 'Well, that's nice, but we've got this other thing' and you know, so it leads to work one way or another is the idea."

- Making use of specialized tools for technical communication within a specialized industry. This included using specialized software for production and project management (e.g., Jira for project management or Mantis for bug tracking) as well as using normal productivity software in specialized ways for various things (e.g., using MS-Excel for recording character dialogue in a level design document).

In addition to our quantified analysis of code frequency, Gee's (2014) Making Strange tool pointed us toward other examples of strategies that worked against what we might expect in a development studio. For example, Taylor described how e-mail, which we suspected would be widely used to facilitate problem solving and task distribution, was not a preferred method for communication for many developers:

Yeah, e-mail's never been big at n-Space. Face to face is good. Anything important I always write things down for myself. If we'll have a meeting, there's usually 20 or 30 things or a bunch of people chat in a room ... a lot of that detail, if it doesn't get written down, it gets lost. Or half of it gets lost or somebody mangles it in their brain so some feature gets put into the game that's not quite what we talked about. That sort of thing. So what I like to do is whenever we have these face-to-face meetings I always carry a notepad around with me. Write these things down and then I try to do a recap at the end of meeting that say okay this is what we talked about, blah blah blah, and then if people already have their own notes, I usually send out a recap meeting. And then hopefully a lot of that stuff gets put into tasks like we use within Jira.

This example worked against our assumptions and provided new information that helped "make strange" the discourse of independent game development. It emerged from a specific challenge, the tendency for vital information to become lost in translation, ignored, or corrupted if not written down. This too was interesting, because e-mail would provide another method for writing things down, yet it does not appear to be a preferred tactic for employees. Finally, it points out particular tools (e-mail, a notebook, Jira), texts (the content Taylor has written down in the notebook, each individual attendee's notes, the task lists logged into Jira), and tactics (sending out a request for a recap meeting) that illustrate how Taylor has used technical communication strategies to overcome these constraints.

Conclusion

This research explored the discourse and culture of an independent game studio by analyzing the tools, tactics, and texts that emerged from major conflicts and constraints within the video game development environment. Interviewed participants spoke about practices that seemed less like "dirty tricks" supporting a system of "smoke and mirrors" and more like smart communication practices that recognized diverse audiences and applied hybrid design practices. Understanding how an organization deals with problems helps us to understand its culture. This knowledge, as we argued in the introduction, is important for the educators of future technical communicators who will work in such environments as well as for practitioners who can add value to existing communication practices inside game studios. Such individuals may also transfer communicative expertise from game development to other technical production environments.

The employees at n-Space radiated professional ethos and competency. Many interviews reflected attitudes of pride and ownership of working in an independent studio rather than for a larger commercial development company. However, such pride also pointed to the reality of working as a small studio with access to fewer resources than larger companies. These references spoke to the embedded cultural values of n-Space. Bryce explained:

We're the dark horse. We've always been labeled the dark horse or the underdog you know and but you look and we're here almost twenty years later standing in an industry where some of the titans and much bigger people than us have fallen.

Thus one central conflict of n-Space—its positioning as an underdog against larger developers—introduced constraints, but also certain benefits. Culturally speaking, such an ethos of scrappiness was valued by employees such as Bryce, and operationally speaking, a nimbleness and flexibility was possible in the studio that may not have been in larger, more compartmentalized companies.

Future research should continue to investigate a number of understudied areas within the intersection of game development and technical communication. One area ripe for analysis within the present data set is studying the relationship between strategic, top-down administrative strategies and the actual practices of studio development employees. Our analysis has not drilled down into communication style or content differences between studio administration and lower level employees, for example, which would highlight areas of effectiveness or misalignment between studio mission and actual practices. Additionally, we need to better understand the cultural components of game development, particularly in relation to building support for more workforce diversity. More research in these areas will allow us to understand further the discursive strategies of professional game developers.

Notes on contributors

Rudy McDaniel is associate professor of digital media at the University of Central Florida. His current research focuses on knowledge management and communication strategies in video game development. He also researches digital badges.

Alice Daer is a content strategist at Findlaw, a part of Thomson Reuters. She is also a member of SIGDOC, the Association for Computing Machinery's special interest group on the design of communication, and serves on the editorial board of its flagship journal, *Communication Design Quarterly*.

ORCID

Rudy McDaniel  <http://orcid.org/0000-0002-9729-111X>

References

- Cohendet, P., & Simon, L. (2007). Playing across the playground: Paradoxes of knowledge creation in the videogame firm. *Journal of Organizational Behavior*, 28(5), 587–605. doi:10.1002/job.460
- Daer, A. J. (2010). This is how we do it: A glimpse at Gamelab's design process. *E-learning and Digital Media*, 7(1), 108–119. doi:10.2304/elea.2010.7.1.108
- deWinter, J., & Moeller, R. M. (Eds.). (2014). *Computer games and technical communication: Critical methods and applications at the intersection*. Burlington, VT: Ashgate.
- Eyman, D. (2008). Computer gaming and technical communication. *Technical Communication*, 55(3), 242–250.
- Gee, J. P. (2014). *How to do discourse analysis: A toolkit*. New York, NY: Routledge.
- 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. In J. deWinter & R. M. Moeller (Eds.), *Computer games and technical communication: Critical methods and applications at the intersection* (pp. 17–33). Burlington, VT: Ashgate.
- Lamberti, A. P., & Richards, A. R. (2012). Gaming/writing and evolving forms of rhetorical awareness potentials of interactive digital media for democratic classrooms. *Pedagogy*, 12(3), 481–495. doi:10.1215/15314200-1625262
- Mason, J. (2013). Professional writing and video games. *Connexions: International Professional Communication Journal*, 1(1), 173–178.
- McAllister, K. (2004). *Game work: Language, power, and computer game culture*. Tuscaloosa, AL: University of Alabama Press.
- McDaniel, R. (2009). Making the most of interactivity online version 2.0: Technical communication as procedural architecture. *Technical Communication*, 56(4), 370–386.
- n-Space. (2015). *Official web site*. Retrieved from <http://n-space.com/>

- O'Donnell, C. (2009). The everyday lives of video game developers: Experimentally understanding underlying systems/structures. *Transformative Works and Cultures*, 2. doi:10.3983/twc.2009.0073
- Pan, S. L., & Scarbrough, H. (1999). Knowledge management in practice: An exploratory case study. *Technology Analysis & Strategic Management*, 11(3), 359–374. doi:10.1080/095373299107401
- Robison, A. J. (2008). The design is the game: Writing games, teaching writing. *Computers and Composition*, 25(3), 359–370. doi:10.1016/j.compcom.2008.04.006
- Saldaña, J. (2013). *The coding manual for qualitative researchers* (2nd ed.). Thousand Oaks, CA: Sage.
- Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: MIT Press.
- Seidman, I. (2013). *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. New York, NY: Teachers College Press.
- Spinuzzi, C. (2015). *All edge: Inside the new workplace networks*. Chicago, IL: The University of Chicago Press.

Appendix: Survey questions

Non-Narrative Prompts

- What is your name and job title?
- How do you identify yourself racially/ethnically/culturally?
- What group or division do you work in and who is your direct supervisor?
- What are some of your primary job responsibilities?
- What activities do you perform in an average day?
- What sources of data are essential to your position?
- What sources of data do you personally generate?
- What constraints do you have to deal with on a regular basis?
- What communication strategies do you use to interact with other employees?
- Who are the typical audiences/clients/customers you engage with or interact with?
- How do you communicate with your direct supervisor?
- How is technology used by your organization and your stakeholders/customers?
- What is the primary mission of your organization, as you understand it?
- Which three adjectives would you use to describe yourself as an employee?
- How do you evaluate successful communication in your environment?

Narrative Prompts

- If you have one, describe an experience or tell me a story ...
- ... about a failure you experienced or a situation in which you did not perform as well as you would have liked.

What did you learn from that experience?

- ... about a situation in which communications between you and other individuals did not go as you expected.
- ... about conflict. What happened, and how did you deal with it?
- ... about a situation in which you had to seek help from others.
- ... in which you had trouble locating the knowledge necessary to achieve a specific goal or objective.
- ... in which you believe that cultural factors prompted an audience or stakeholder of one of your communications to interpret your message differently than you originally intended.

At this time, I'd like you to share any additional interesting stories or memorable moments, of your choosing, that provides insight as to what it is like to do your job, or provide insight into the knowledge management or communication practices of this organization.