Author: **Beecken, Kathryn M.**

Title: ***Genre Analysis of Software Release Notes***

The accompanying research report is submitted to the **University of Wisconsin-Stout**, **Graduate School** in partial completion of the requirements for the

Graduate Degree/ Major: Technical and Professional Communication MS Degree

Research Advisor: **Gregory Schneider-Bateman, Associate Professor**

Submission Term/Year: **Spring 2019**

Number of Pages: **76**

Style Manual Used: **American Psychological Association, 6th edition**

X **I have adhered to the Graduate School Research Guide and have proofread my work.**

X **I understand that this research report must be officially approved by the Graduate School. Additionally, by signing and submitting this form, I (the author(s) or copyright owner) grant the University of Wisconsin-Stout the non-exclusive right to reproduce, translate, and/or distribute this submission (including abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video. If my research includes proprietary information, an agreement has been made between myself, the company, and the University to submit a thesis that meets course-specific learning outcomes and CAN be published. There will be no exceptions to this permission.**

X **I attest that the research report is my original work (that any copyrightable materials have been used with the permission of the original authors), and as such, it is automatically protected by the laws, rules, and regulations of the U.S. Copyright Office.**

X **My research advisor has approved the content and quality of this paper.**

STUDENT:

NAME: Kathryn Beecken **DATE:** **April 30, 2019**

ADVISOR: (Committee Chair if MS Plan A or EdS Thesis or Field Project/Problem):

NAME: Dr. Gregory Schneider-Bateman **DATE:** **April 30, 2019**

This section for MS Plan A Thesis or EdS Thesis/Field Project papers only

Committee members (other than your advisor who is listed in the section above)

1. **CMTE MEMBER'S NAME:** **DATE:**
2. **CMTE MEMBER'S NAME:** **DATE:**
3. **CMTE MEMBER'S NAME:** **DATE:**

This section to be completed by the Graduate School

This final research report has been approved by the Graduate School.

Director, Office of Graduate Studies: **DATE:**

Beecken, Kathryn M. ***Genre Analysis of Software Release Notes***

Abstract

This study performs a corpus-based genre analysis of software release notes. It begins by evaluating the context and communicative purpose of software release notes that are used by technical teams to upgrade organizational software systems. Based on the framework of how release notes are used, a close examination of release notes in the corpus identifies a common structure for how a collection of release notes is presented and the moves within individual release notes. Commonly used textual features, such as bulleted lists, bold text, and hyperlinks, are also seen throughout the corpus.

In addition, several features are observed that are unique to the role of release notes and are not seen in traditional instruction manuals, such as types of notes specific to a point in time and intended for a different audience than the end users. These findings help technical communicators who write release notes adhere to emerging conventions of the subgenre and pave the way for future studies into release notes. Recognizing release notes as a distinctive form of technical communication demonstrates the need for further study and training of release note writing.

Table of Contents

List of Figures

Chapter I: Introduction

Organizations need to regularly upgrade their software infrastructure, and release notes are a key player in software upgrades (Morgan & Ngwenyama, 2015). Software companies publish release notes with tasks that organizations need to complete as part of upgrading to a new version of the software. Release notes are an increasingly prevalent and essential subgenre of technical communication, serving a very specific role yet impacting a wide array of industries. This study conducts a genre analysis of software release notes to understand common attributes that release notes share and which attributes are unique to this subgenre to accomplish its communicative purpose.

Release notes warrant examination because of their widespread use, significant impact, and distinctive purpose. Software is the backbone of operations in nearly every industry. From manufacturing to sales, healthcare to education, software plays a significant role from daily user actions to overall organizational profitability (Barth, 2017). On the back end, IT teams manage and maintain software systems to ensure the software meets the needs of their respective organizations. Software companies use release notes to communicate to IT teams what actions they need to take at the time of upgrade to successfully use the newest version of the software. In this ecosystem, nearly all organizations use software, software requires frequent upgrades, and release notes are part of the path to upgrade. Consequently, release notes are increasingly prevalent and used by many organizations across industries.

In addition to being widespread, release notes are critical because of their impact on software systems and organizational operations. When an organization upgrades their software system, they need to be able to do it with minimal disruption to ongoing operations (Morgan & Ngwenyama, 2015). From the perspective of the software company, releasing a new software version is a high-risk activity, in part because the IT team at each organization that uses that software needs to be able to effectively upgrade to the new version. A study of a provider of IT services identified improved release notes with a shared language between developers and consumers as a possible solution to smoother software upgrades (Jokela & Jantti, 2012). As the key mechanism for communicating upgrade steps to technical teams, release notes directly impact the success of software upgrades and system function.

Because of this specific use case and audience, software release notes are a unique subgenre. While bearing some resemblance to traditional user instructional documentation, they are distinct in the technical nature of the content and the specialization of the audience, which necessitates different guidelines for content and format (Phoha, 1997). Also, they are used at a specific point in time when an organization that is already using the software is upgrading to a new version of the software, in contrast to installation manuals that are used for initial installation. Instead of telling the entire story of a product from beginning to end, software release notes are much more targeted in scope to changes and tasks that are relevant to upgrade.

Statement of the Problem

As a subgenre, software release notes merit closer examination because of both their ubiquity and impact. This leads to the following research questions:

- RQ1 - What similarities unite software release notes as a subgenre?
- RQ2 - What attributes are unique to release notes as part of accomplishing their communicative purpose?

Purpose of the Study

By answering these questions, practitioners who write release notes for software companies will be better equipped to understand and embrace attributes of the genre. Also, educators will be able to train technical writers specifically on release note writing, which can ultimately lead to stronger release notes and improvements to the software upgrade process.

Assumptions and Background of the Study

A wide range of documentation can be categorized as release notes, from a list of changes that you see when upgrading an app on your phone to comments within program code itself and intended for other developers. This study examines release notes for software systems that are read by system analysts or administrators who are upgrading the software that is used by end users. This focus was chosen because these release notes are the most critical in terms of company success in that they require action and have consequences across the organization.

Because this type of release note is behind the scenes and unfamiliar to most general audiences, the following example illustrates the role of release notes and the actors involved:

1. Academic Institution I uses learning management system software created by Company A.
2. Company A releases a new version of the software and encourages all of the academic institutions that use the software to upgrade to the new version to get several fixes and enhancements to the software.
3. Company A publishes release notes that describes the tasks that analysts need to complete to upgrade to the new version of the software.
4. IT analysts at Academic Institution I read the release notes published by Company A and take the necessary actions to upgrade to the new version.
5. Instructors, students, and other users of the software at Academic Institution I use the new version of the software.

In an ideal state, the transition for end users (instructors and students in the previous example) is seamless and non-disruptive, while providing improvements to the software. However, if something goes wrong with the software upgrade, users might experience frustration with the system, such as inability to perform certain actions or even system outage. Release notes are a key vehicle for assisting IT analysts or system administrators in a smooth upgrade process and the medium through which the software company communicates to the analysts managing the software.

Limitations of the Study

In light of the wide variety of release notes and the vast array of software companies that use different forms of release notes, this study is limited at looking at a small set of release notes within narrow parameters. This study is also limited by the lack of data for gauging the effectiveness of individual release note attributes or the relative success of the release notes within the corpus. Without this outcome data, this study can make observations about similarities between release notes and differences from other forms of technical writing to answer the research questions, but it is unable to offer a qualitative assessment or recommend best practices.

Methodology

This genre analysis of software release notes begins with a literature review that surveys common methodologies for genre analysis and the role of genre analysis in understanding types of documentation. The literature review also looks at the limited previous research about release notes.

To answer the research questions, this study analyzes a sample of release notes from six different software companies in six different industries. Using an adaption of the Bhatia method for genre analysis, the communicative purpose is identified and the structure, textual features, and writing style that support that purpose are analyzed. Based on this examination, the study concludes with a discussion of the findings and recommendations for further research. More information about the corpus of release notes studied is found in the Appendix.

Chapter II: Literature Review

Previous literature on genre analysis shows common methodologies for evaluating a genre as well as differing perspectives in how genres are understood. The second section in this literature review looks at previous studies about software release notes, although few studies have so far investigated release notes from the point of view of the audience, and none through the lens of genre analysis. Reviewing these two tenets of previous study reveals a gap where a genre analysis of release notes has not been done and provides a framework for the study.

Genre Analysis

In *Modern Genre Theory,* genre is defined as "a recurring type or category of text, as defined by structural, thematic, and/or functional criteria" (Duff, 2014). Hyon (2017) defined genre analysis as "the study of genres and their contexts," although the simplicity of the definition falls short of capturing how genre analysis has emerged as a key component of rhetorical studies and become a popular lens for communication studies. Hyon's definition also avoids the debate about whether to use a textual approach or a social context-based approach for genre analysis (Luzon, 2005). As genre analysis is applied to an increasingly large range of texts to understand rhetorical discourse and educate communicators about their specific disciplines, these two approaches have drawn closer together, although differences remain in methodology and emphasis.

Genre analysis using a textual approach. Aptly named, a textual approach to genre analysis focuses on linguistic characteristics of the text (Luzon, 2005). In his exploration of the development and application of genre analysis, Hyon (2017) hailed John Swales as the founding father of the field. Swales (1990) conducted a groundbreaking genre analysis of research articles that attempted to answer how language is used for the communicative purpose of a specific genre. He developed the move-step method to identify the overall communicative purpose of the text, the moves that each have an individual purpose while contributing to the overall purpose, and the steps that actualize the moves (Swales, 1990).

This approach has been used for a wide range of studies and adopted by English for Specific Purposes (ESP). In their study of tax computation letters, Flowerdew and Wan (2006) used this method at both the macro-textual level to analyze the moves and frequency of moves and at the micro-textual level to look at linguistic choices such as politeness, mood, and modality. Parkinson (2017) used a similar approach in his study of student laboratory reports to identify the function of each text segment, which he thematically grouped and coded as moves and steps. Ding (2007) provided another example in his analysis of application essays to medical and dental schools where he began by defining the communicative purpose of the genre and then identified moves based on textual divisions and linguistic clues. He also used software to identify the frequency of keywords and how they related to the moves (Ding, 2007). Perhaps most similar to this study, Frutos (2015) used Swales move-step method in a corpus-based genre analysis contrasting instruction manuals for household appliances written in English and in Spanish. This sampling from the scores of genre analysis studies that use a textual approach, and specifically the Swales move-step methodology, show its application to a variety of texts and its validity as a strategy for understanding a genre by looking closely at the linguistic features employed to achieve the communicative purpose of the genre.

Genre analysis using a social-context approach. In contrast to focusing on the specific language of a text, the Rhetorical Genre Studies (RGS) approach examines the social purpose (Hyon, 2017). This school of thought traces its origin to Miller's paper on how a genre is defined by the action accomplished and not by the form of the text (Miller, 1984). While a scholar using a textual method would begin with the communicative purpose of a text and then dig into line-level details to determine how the text accomplishes the purpose, the social-context approach takes a step back to see genre as a social institution and understand it through the perspective of the discourse community.

With its wide-angle view, the social-context approach has been applied to many situations to understand the people, places, and cultural factors that can determine rhetorical choices. Yates and Orilkowski have conducted multiple studies in which they define genre as having a socially-constructed purpose that is recognized by the relevant organizational community, instead of the individual motive of the author (2002). In their study of professional collaboration in an online chat room, Yates and Orilkowski (2002) identified that more than half of the communication adhered to three overlapping genre systems that had previously been established. Despite the uniqueness of the space and its relative novelty, the discourse in the chat room still served similar social goals to existing genres and thus could be considered part of those genres. Artemeva (2005) applied a social context approach in her longitudinal case study of how an engineer imported genre conventions from a previous workplace into a new workplace, showing how previous experiences could transfer best practices to shape rhetoric in a new context. By focusing on the human aspect of how and why people communicate, the social-context approach to genre analysis can have a substantially different angle on understanding a genre.

Genre analysis using a hybrid approach. The social-context approach and textual approach have increasingly influenced each other, and many recent studies use a hybrid methodology that focuses on textual characteristics while keeping an eye toward the larger social context of the genre. For example, Rutherford (2005) conducts a linguistic study of corporate annual report narratives using word frequency and word lists. Despite his textual methodology, he uses Miller's definition of genre and draws from Yates and Orilkowski to understand the social factors that could affect the corporate annual report narratives, such as the state of the market, company size, and company profitability (Rutherford, 2005).

Bhatia's foundational work *Analyzing genre: Language use in professional settings* (1993) outlined a seven-step methodology for genre analysis. Before beginning textual analysis, multiple steps look at social and contextual influences on a genre. Since his initial publication, Bhatia's methodology has been revised and adapted numerous times, by both himself and others, to answer the fundamental question "why do professionals in this particular field use language the way that they do?" (Bhatia, 2008). He described his method as including both text-external information, or the social conventions that make a genre possible, and text-internal features at the linguistic level (Bhatia, 2008). Singh, Shamsudin, and Zaid (2012) used Bhatia's method in their genre analysis of workplace documents produced by Malaysian engineers in the petroleum industry. They concluded that the combined macro and micro elements in Bhatia's approach provided a robust, multidimensional analysis of the genre (Singh et al., 2012). Zhou (2012) used a modified version of Bhatia's seven steps in a genre analysis of advertorials, and this paper follows a similar structure to that study.

Based on the precedent for using both the textual and social context methods and the benefits of a hybrid approach, this study uses a modified version of Bhatia's seven-step method to understand the genre of software release notes. This method is further described in the methodology section.

Software Release Notes

Despite the increasing role of release notes as a tool for software upgrades, few studies to date have analyzed release notes as a form of rhetoric. While at first blush, release notes might appear similar to traditional instructional documentation, release notes are a distinct subgenre with a specific audience of IT or system administrator team. In contrast to manuals for consumer goods that provide operating instructions for users, release notes provide setup instructions for an audience other than the end user. Because of the specific context where release notes are used, they require separate examination. Previous studies have looked at how release notes are created or generated, but there has been little analysis of the end product and content of the release note.

Previous studies of release notes for developers. Several studies have looked not at release notes but at change descriptions that are intended for other developers. These change descriptions are different from the release notes that are the subject of this study because software developers have a different responsibility and level of expertise than the software release notes than the release notes read by analysts to upgrade an organization's software. Dekel and Herbsleb (2009) conducted one such developer-facing study by using the eMoose documentation framework to guide 26 developers through five tasks to investigate the effectiveness of code-level change documentation for developers. Ding, Liang, Tang, and Van Vilet (2014) also looked at effectiveness for a developer audience through their systematic literature review of 60 studies, which resulted in defining 12 quality attributes of software documentation, four cost categories, and nine benefit categories.

Previous studies about release note creation. Other previous studies have focused primarily on how release notes are created. Buse and Weimer (2010) studied changelog messages that were automatically generated to summarize differences in software code. While quantitatively the log messages that were automatically produced contained 89 percent of the messages in human-produced logs, the general consensus of the developers surveyed was that they wanted the automatic change descriptions as a supplemental resource but not as a replacement for human documentation (Buse & Weimer, 2010).

Abebe, Nasir, and Hassan (2016) studied 85 release notes from 15 software companies in their attempt to develop a machine learning model to automatically determine which changes to include in release notes. Moreno, et al. (2016) further built on this research in their analysis of 990 release notes from 55 software companies to test their tool ARENA to automatically generate change histories by comparing current and previous versions of the software code. Klepper, Krusche, and Bruegge (2016) also focused on accelerating release note creation, but they recognized that different audiences require different information for a software upgrade. They included human review by an upgrade manager to curate the automatically generated list of changes to accommodate the different needs of developers and analysts implementing the upgrade.

While these studies begin to recognize release notes as a distinct genre of documentation, they have limited relevance to the research questions in this study because they are looking at release notes that are used for a different purpose and by a different audience. Also, they are specifically looking at automatically generating changelogs as opposed to analyzing the final content of release notes that require human action.

This review of the literature shows the value in a genre analysis to understand both the textual features and social context of a genre, but that software release notes have not yet undergone this level of study.

Chapter III: Methodology

This study uses an adapted version of Bhatia's seven-step model for genre analysis:

- Step 1: Overview of Situational Context. This context is briefly outlined in the introduction to this study by describing the reason for study and defining the scope of the study.
- Step 2: Literature Review. This is covered in the preceding section by looking at previous studies on software release notes.
- Step 3: Refining the Contextual Analysis. The context for release notes is more thoroughly discussed in this section.
- Step 4: Selecting the Corpus. This step defines how the corpus was selected.
- Step 5: Institutional Context. As defined by Bhatia, this step examines particular rules for the documents studied, such as a company-specific style guide. Because the release notes in this study come from different companies and industries, this study looks rather at how release notes share a communicative purpose, which helps show contextual commonalities that are shared between release notes.
- Step 6: Levels of Linguistic Analysis. This step is the main thrust of the study and the findings below include a linguistic analysis of both release notes collectively and individually.

Bhatia's seventh step for expert review is not included in this study, as there is precedent for not doing expert review for genres where the author has extensive experience or if it is beyond the scope of the study (Zhou, 2012).

Selecting the Corpus

For this study, the corpus selection criteria are derived from the situational context. To be included in this study, the software release notes must meet all of the following criteria:

- Intended to be read by a person who is not the end user of the software.
- Not intended for people who contribute to the development of the software.
- Include one or more required setup tasks (i.e. not informational-only change descriptions).
- Publicly available online.

The release notes must also be published discretely as notes that are distinguished from each other and from other setup documentation. For example, installation guides do not meet the criteria for this study because they are intended for installing users instead of current users who are upgrading to a new version. While installation guides might have flags for changes that are applicable for a certain version only, they are not discrete release notes.

The release notes included in this study represent companies in six different industries: hospitality, restaurant service, education, security, business, and telecommunications. This variety was intentionally chosen to represent a range of fields and neutralize trends pertinent to only a given industry.

Data Collection Procedures

For each of the six companies identified, the release notes for a single product were examined. The analysis included all of the release notes for the last three versions of that product that were released, from December 2018 and earlier. For more details about the specific company and product release notes analyzed, refer to the Appendix. In total, 259 release notes from six companies were included in this study.

The delivery mechanism and presentation of the release notes was first observed at the macro level. This included documenting how the intended audience would access the release notes and observations about their presentation that could influence release note content or user perception.

Then, the release notes were read and qualitative observations were documented about the language, structure, and form of the release notes as a whole for a given product version. The first five release notes for each product and version were more closely analyzed using the Swales method and broken down into moves and steps to understand the structure of individual release notes.

Data Analysis

After concluding the qualitative analysis of the language and structure of the release notes, the notes were quantitatively analyzed based on several metrics. Word count, Flesch-Kincaid Reading Ease, and Flesch Grading Level were calculated using Microsoft Word tools. The number of release notes that contained images or setup instructions were manually tallied. Data analysis was performed using native Microsoft Excel tools.

Limitations

This study examined release notes produced by six companies, which might not be generalizable or applicable across the entire field of release notes. Also, this study focused on release notes as narrowly defined in the scope of study and does not account for other types of release notes outside of the parameters of the corpus. While the release note genre continues to change and develop in response to rapid changes in software technology, the results of this study capture a moment in time for release notes published in the latter half of 2018.

Chapter IV: Results – Release Notes Collectively

The first line of analysis looks across release notes to understand more about the context, the types of release notes, and how collections of release notes are published as a single deliverable. Later results sections look more closely at characteristics specific to individual release notes.

Context and Communicative Purpose

Knowing more about the context and communicative purpose of both the author and the audience for release notes informs the later analysis of why individual release notes have specific characteristics. Identifying the "who" of release notes and their perspectives on why they use release notes creates the landscape for understanding how release notes function.

It is essential to look from both the perspective of the author and of the audience because they could have different views about the role of release notes. Authors have a conceptual model of their audience that shapes their work but might not be reflective of the actual audience (Coney, 1992). A discrepancy can exist between the purpose of the documentation as seen by the author and the purpose seen by the reader (Coney, 1992). Understanding both viewpoints is necessary for a comprehensive understanding of the purpose of release notes.

Purpose of the author. Release note authors are technical writers or other employees of the software company who work with research and development teams to document software changes. Three of the companies studied stated their communicative purpose in the introductory material for the release notes:

- Cisco: "These release notes describe the new features, improvements and changes in release 2.3 of the Cisco Meeting Server software." (Cisco Meeting Server release notes, n.d.)

- Clock: “Product news - We constantly improve our software, following our development plan and your feedback. See the news in each update.” (“Software update release notes,” n.d.)
- Jamf Pro: “See what's new, key features, and installation guidelines in Jamf Pro release notes.” (“Jamf Pro release notes,” n.d.)

Of these three statements, Cisco and Clock both describe the changes as “improvements,” and Cisco and Jamf Pro both specifically mention “features.” The Clock statement differs from the other two by focusing on the relationship between the company and the customer organization with personal pronouns and highlighting “your feedback.” It also frames the release notes as “news,” which underscores their release notes as dynamically changing information instead of a static instruction manual. These statements show how the software companies producing release notes articulate their perspective of their communicative purpose.

Purpose of the audience. In contrast to examining the purpose of the author, the study must also cross the divide and look at the purpose as viewed by the audience. The organization who uses the software has an IT team of analysts, computer support, or system administrators who are responsible for upgrading the software, which depends on reading and acting on the release notes produced by the software company. Job descriptions provide a window into how organizations that hire people to read and act on release notes perceive them. While job descriptions are an imperfect proxy for hearing directly from the audience, they still provide a glimpse at the audience’s perspective without being mediated by the authors’ assumption of them. Consider the following job postings:

- Federal Reserve Bank (MN): “Technical Systems Analyst - Plan and execute package deployments to computers. Coordinates scripting and packaging of software deployment updates via Jamf Pro and BigFix.” (Monster Jobs, n.d.)
- CUNY Queens College (NY): “I.T. Associate, Level 1 - Blackboard administrator: deliver ongoing support such as troubleshooting complex user problems and will work closely with the Enterprise Applications Services team to provide back end software development.” (Monster Jobs, n.d.)

From the first job description, it appears that technical systems analysts are responsible for Jamf Pro updates, and presumably release notes. They have the technical knowledge to script and deploy software packages, and they are also responsible for multiple software systems, or specifically BigFix in this case. The second job description indicates that a person reading Blackboard release notes works both with end users and with other technical teams. Their purpose is software support, including troubleshooting and development. Gleaning this information about the audience from job descriptions provides a foundation for evaluating how specific attributes of release notes reach their audience, such as in level of complexity and assumed system knowledge.

Other job roles that are responsible for reading the release notes produced by the specific companies included in this study ranged from a System Engineer job posting for Expedia (WA) that requires knowledge and experience with Jamf Pro to an Instructional Designer job posting for Finger Lakes Community College (NY) that requires performing administrative tasks for Blackboard (Monster Jobs, n.d.). The range of positions that are expected to read the release notes in the corpus shows that the audience is not homogenous but rather represents a variety of needs and contexts. Yet it also highlights commonalities among readers, such as their responsibility for system support and their technical background.

Comparing author and audience purpose. Comparing company purpose statements with organization job descriptions reveals both similarities and differences in how authors and readers perceive the communicative purpose of release notes. From the view of the software company, the release notes highlight improvements to the software and provides the audience with updates. From the view of the organization that is hiring an analyst to read release notes and upgrade their software system, the purpose of release notes is software support, and analysts are expected to support users by upgrading the software, which entails processing release notes. These purposes are similar in that both are focused on software support and that is clearly the reason why release notes exist and why they are written and read.

However, the views differ in how the author and the audience perceive the scope of impact of the release notes. The organization that is reading release notes has a broader perspective of overall deployment and user experience, but the software company is specifically focused on how new software versions make improvements. Part of the reason for this gap might be that job descriptions are a lot more general and encompass more than just release note review, whereas the purpose stated by the software companies is included with release note documentation and very specific to release notes. Also, for the software company, the primary object is the software system. On the webpage or document where the software company publishes release notes, the focus is the release note itself, in support of the larger system. From the audience perspective, the primary focus is not the actual release note. The software is merely a means to an ends and in support of their larger service, such as Clock supporting the ability to manage a hotel chain, and release notes are even one more step removed. For the audience, the software supports their business, and the release notes support the software, but the release notes themselves are not the focus.

There is also a difference in tone, where the job postings present the work as routine, but the software company is framing the changes in the new version positively, such as in Clock's language "We constantly improve" ("Software update release notes," n.d.). This difference in tone is not surprising considering their respective roles as sellers and buyers of the software and the marketplace dynamic. Ultimately, the same underlying purpose of upgrading the software drives both the software company and the organization and is the communicative purpose of release notes but with different perspectives on the amount of focus on the release notes and in their tone.

Types of Release Notes

The next area of analysis when looking across release notes is understanding the different kinds of release notes. Examining the "what" of release notes also shows how they differ from traditional instruction manuals.

Linguistic tradition of instruction manuals. Release notes are new with the birth of software and the software upgrades that necessitate them, but their lineage can be traced back to traditional instruction manuals. Consider traditional instruction manuals in two categories:

- Installation manuals. These manuals are used to set up or begin using a product for the first time and are typically not referred back to once the product is in use.
- User manuals. These manuals provide ongoing instructions for the use of the product, and the user might refer to them to initially learn how to use the product or to answer questions after the product has been used for a while.

In a pre-software age, installation manuals were physical documents that accompanied a new product. Software usually has equivalent installation manuals for how to set up a product for the first time, even if the instruction manual document is now rendered electronically, such as a PDF, instead of delivered as a hard copy. For example, when an organization is implementing Cisco Meeting Server for the first time, they begin with the Cisco Meeting Server with Cisco Expressway Deployment Guide, or another deployment guide depending on their deployment method. The deployment guides include a basic introduction, terminology, an overview of installation options, and how to set up each part of the software for the first time ("Cisco Meeting Server with Cisco Expressway," 2016).

Release notes are similar to installation manuals in that they are used to install the new version of the software and are then discarded. However, release notes are quite different than installation manuals because of their unique position to upgrade a product that the user is already familiar with. They represent a specific point in time to take software from one version to the next, and they are not intended to be a comprehensive body of information about the software. Reading the Cisco Meeting Server release notes would not help an organization implement the software for the first time because the release notes are targeted change descriptions instead of an entire setup guide. This fundamental difference in timing and purpose is seen in the types of release notes observed in the corpus.

In the software industry, there are also often user manuals that serve a similar purpose to traditional user manuals by helping users interact with the software. When a person first begins using a product, they read the user manual to learn how to use it, and if they later encounter questions about a given aspect of use, they can refer back to the user manual. For example, the Blackboard website includes a Learn page with resources specifically designed for students and instructors (“Blackboard Learn,” n.d.). These user manuals teach the end user of the software how to use the various features and functions to accomplish the desired tasks.

In a sense, release notes are similar to user manuals because they are about a product that the user is assumed to already have set up. However, release notes expect users to have a level of familiarity with the previous version of the software, and user manuals might be for beginning users who are being exposed to the software for the first time. Another key difference is in audience because the release notes that were analyzed for this study were specifically chosen as being intended for an IT analyst team that is separate from the end user of the software. While user manuals are intended for the user, release notes are typically for set up behind the scenes.

For many products, the same person is both doing the initial installation and is the end user of the product. The installation manual and the user manual come together, perhaps even as different sections within the same document. For example, the instruction manual for the KitchenAid Stand Mixer has one section entitled “Assembling Your Tilt-Head Stand Mixer,” which is an installation step, and the next section is “Using the Pouring Shield,” which is a use step (“Instructions and recipes for your KitchenAid stand mixer,” 2009). The same can be true for consumer software products, such as including instructions for both how to install the software and how to use it.

Fixes and enhancements. Broadly speaking, all release notes could be categorized as describing a either a fix or an enhancement. Fix release notes describe how to correct broken functionality and restore intended behavior. For example, Blackboard published the following release note for version 9.1: “Security Fix - This release contains a security fix to the login page that could interrupt the way third party SSO solutions pass user credentials from a client portal to Blackboard…” (“Release notes for Blackboard Learn,” n.d.).

The vast majority of the release notes analyzed were enhancements because they introduce a new feature or improve system behavior. For example, Accounting Seed published the following release note for the Echinacea version: “Enhancement - Multi-currency support for Bank Deposits. A currency can now be selected when creating a bank deposit and this restricts available Cash Receipts and Journal Entry Lines to the currency entered” (“Release notes,” n.d.).

In both of these examples, the release notes are explicitly labeled as a fix or enhancement by the company. However, the line between fix and enhancement can be blurred, and changes could be described in such a way to make them sound like either. For example, Blackboard could have written the “security fix” release note above instead as a “security enhancement” because the system is now more secure in the situation described. Likewise, the Accounting Seed change could instead be seen as a fix if the lack of multi-currency support for bank deposits was perceived as deficiency that has now been fixed.

For both fixes and enhancements, the release notes assume that the user has a starting point with the software, which demonstrates a difference from instruction manuals. A fix makes sense only if the user has experienced the incorrect behavior, and an enhancement has meaning compared to the previous functionality that has now been improved. The example of the Blackboard security fix applies only to an audience that is already using the Blackboard log in and were subject to this vulnerability (“Release notes for Blackboard Learn,” n.d.). The audience that needs to act on the Accounting Seed enhancement is organizations that are already using bank deposit functionality and now need to complete setup to enable multi-currency support (“Release notes,” n.d.).

In contrast to release notes, installation manuals tell the user how to implement a product for the first time and user manuals describe how to use a static product. They do not describe a fix or an enhancement to the product but rather the baseline state of the product for a user that is not previously familiar with it. For example, the Bookcase (5925) – Assembly and Operation Manual published by MDB Family (2015) is an installation manual that lists the hardware and gives assembly instructions for how to put together the bookcase for the first time. It would not be meaningful for MDB Family to include fix or enhancement information. To use a hypothetical example, the reader does not need to know that this model of the bookcase has been fixed or enhanced to be more stable than a previous model by lengthening the screw size because the audience is unlikely to have encountered a previous model of the bookcase. The reader is starting from scratch, instead of upgrading a product that they already know. The fact that release notes are presented as fixes or enhancements stems from their unique point-in-time context and sets them apart from installation manuals.

Changes that affect users. Within fixes and enhancements, additional subcategories of changes were observed in the release notes studied. One of these categories is changes to the software that affect end-user interaction, whether functional or visual only. For example, a release note for Blackboard version 9.1 described a new capability for instructors to record verbal feedback, which is a functional change ("Release notes for Blackboard Learn," n.d.). A release note for TouchBistro version 8.1 described an updated layout for order screens, which does not functionally change the behavior of the software but is an aesthetic change to the user interface that end users are likely to notice ("What's new," n.d.).

These functional and visual changes compose a subcategory of release notes because of their effect on end users who are different than the people reading the release note. In comparison, traditional instruction manuals for consumer products are read by the same people who perform the steps. Installation manuals are read by the people who are doing the initial installation, and user manuals are read by the people who are using the software. When the same audience is both assembling and using the product, the instructions come together, as seen in the KitchenAid stand mixer example above. The intent of the MDB Family bookcase manual is that the person who reads the assembly instructions also assembles the bookcase ("Bookcase (5926)," 2015). However, the analyst who reads the release note about the new verbal feedback functionality for Blackboard is not the instructor who will actually be using the feature to give feedback to students.

None of the release notes in the corpus addressed this gap in audience. They cover the installation side of how to set up the product, but they lack a user manual component to teach end users about the change. The release notes are clearly intended for analysts or system administrators, and there are clearly changes within the release notes that would affect user behavior. However, no attempt was apparent within the release notes to facilitate communicating the change to users. In the example of the Blackboard verbal feedback recording feature, the release note provided the appropriate analyst information for how to configure the new feature but did not include end-user facing steps ("Release notes for Blackboard Learn," n.d.). The Blackboard instructor guide serves as a user manual for how to use grading features in Blackboard. It includes a subsection on the verbal feedback recording feature, but it has notably different content than the release note that reflects the different in audience ("Grade assignments and provide feedback," n.d.). Instead of the setup steps for analysts, it rather includes workflow information for instructors about how to use the feature, and it is longer with more screenshots ("Grade assignments and provide feedback," n.d.). However, there is nothing within the instructor guide to indicate that this feature is new with version 9.1. An analyst gets point-in-time notification at upgrade about how to install this new feature through the release note, but the new feature appears to be incorporated into user manuals without any kind of change notification to instructors.

Depending on the industry and organization, it is possible that analysts are responsible for compiling and communicating end-user changes, that a separate software training team handles this communication, or that the changes are considered intuitive enough that no proactive training is necessary. In one example, Northern Illinois University used a post on their Faculty Development and Instructional Design Center blog. The assistant director of the center wrote a blog post to faculty to summarize the major changes that they would notice with the new version of Blackboard, and also linked resources and hosted an online workshop to provide more information (Richter, 2018). However, none of the documentation from the six companies included in this study accounted for this end-user communication as part of their release notes, and a different role at the organization could be responsible for communicating changes to users.

While this split between the installing user and the end user of the software is perhaps more prominent in this category of software release notes, it is worth nothing that a similar distinction in audience can also be seen in traditional instruction manuals. For example, the Defense Contract Management Agency publishes the Covering for Waterborne Main Propulsion Shafting on U.S. Naval Surface Ships and Submarines manual about the standards for installing propeller coverings on ships (2010). However, this installation manual would be read by a different audience than the personnel who are responsible for inspecting and working worth the propellers. These users would instead participate in Naval Propulsion Program Training (“Naval Propulsion Program (NPP) Training,” n.d.), which serves the role of a user manual to teach users how to use the product. This situation is similar to release notes in that the audience for installation is different than the audience for using the product, but this scenario seems relatively rare for traditional instruction manuals, compared to the relative frequency for release notes. Also, the example of the naval propulsion manuals demonstrates successfully providing both installation and user instructions that are targeted to the appropriate audiences. The release notes studied have this gap in audience but fail to provide the equivalent of a user manual for the end user of the software.

Reversing a change. A less common category of release note observed was how to alter or reverse automatic changes. The majority of software release notes included a description of the change and, if needed, steps for how to enable the change or further configure it. However, some release notes were structured the opposite way. These notes described an automatic change that takes effect as soon as the software system is upgraded, and the release note has optional steps to block the automatic change. These steps could be a “kill switch” to put an emergency stop to the change, a way to reverse to the previous behavior, or an option for modifying a new default setting. Instead of giving the audience a choice to opt in to a change by setting it up, these changes are automatic and the release note describes how an organization could opt out.

For example, Cisco version 2.3. release note 2.10-Support for Dual Screen Endpoints Enabled by Default includes a subsection for disabling dual screen endpoint support if the user does not want to adopt the new default behavior (Cisco Meeting Server release notes, n.d.). This type of de-setup instructions occurred occasionally throughout the release notes reviewed and appears to be a unique characteristic of the genre. In traditional installation manuals, such as to build a physical product, nothing is set up automatically, and a user must consciously perform all necessary setup steps. Or, if a user has already made configuration settings, those settings persist without an automatic change to the default behavior. For example, if you set up the KitchenAid stand mixer, nothing automatically changes your setup. In contrast, software upgrades can include automatic changes, and thus necessitate a unique type of instructional documentation on how to reverse these changes or adjust settings if needed.

Beta or preview features. Two of the six companies analyzed included release notes for beta features. These features are not yet fully developed or ready for general adoption, but they are available in a preview or testing mode. For example, organizations might want to try these features if they need them to support a specific workflow that is unique to them or if they are a cutting-edge organization that wants to test the newest features. Information for features that are not yet formally released is a subcategory of release notes that gives users a look ahead at what is coming and the ability to try a future feature at their own risk. Other types of documentation do not seem to have a corollary concept.

GDPR information. The European Union's General Data Protection Regulation (GDPR) went into effect on May 25, 2018. Because of the timing of this sample of release notes being from the latter half of the 2018 calendar year, all of the companies studied had at minimum one release note that addressed changes to comply with GDPR. This trend suggests that another subcategory of release notes might be notes about widespread changes that necessitate individual product follow up. It is unlikely for traditional instruction manuals to include a similar subcategory because they are not change documentation. However, this type of broad change with a corresponding release note across industries might be unique to this sample and not a frequent occurrence even within the release note genre.

Findings About Release Note Collections

After establishing who is writing and reading release notes and what release notes are, the next angle of analysis is how release notes accomplish their communicative purpose. These findings are further divided into collections of release notes and individual release notes.

A company publishes all of the release notes for a given version together as a collection. For example, Cisco published 12 release notes for Meeting Server version 2.3 (Cisco Meeting Server release notes, n.d.). For the purposes of analysis, these 12 release notes are treated as a single unit of documentation to aid organizations in upgrading to Cisco Meeting Server version 2.3. The collection of release notes for each product and version in the corpus are treated the same way. Later sections in this paper discuss findings for how each individual release note is written to contribute to that goal.

How release notes are presented and delivered. The companies analyzed used two delivery mechanisms:

- PDF: Cisco and Accounting Seed publish release notes as PDF documents that are available for download from the company website.
- Web page: Jamf, Blackboard, and Clock publish release notes directly to a page on their website dedicated to changes in the latest releases.
- Combination: TouchBistro publishes both a PDF and a web page version.

Note that several other formats were observed but not included in the corpus because they did not result in discrete release notes. These included patch notifications that were directly linked to issue tickets and version change descriptions that were embedded within other technical documentation.

It was not apparent what mechanism any of the companies used to deliver the documentation to the audience or announce its availability. When an organization purchases a new version of the software or pays a fee to upgrade, analysts access the release notes that they need to complete their responsibility for upgrading the software system. In the “Preparing to Upgrade Jamf Pro” article, step 1 is “Read the Release Notes” (2011). However, it is not clear if users are sent a pointer to the release notes when they begin the upgrade process, receive the release notes when they purchase a new version of the software, or if they must proactively find the release notes on their own from the company website. The context around how users access the release notes could affect their perception of their importance or of how promotional they seem in nature.

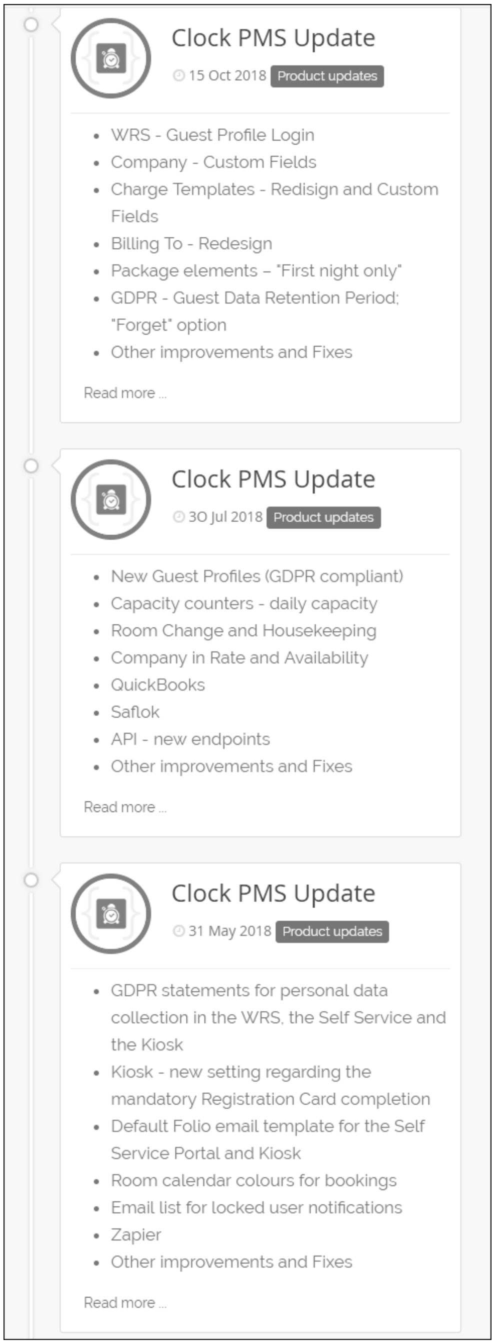
Five out of the six companies studied had a brief, high-level summary that preceded the more detailed release notes. For example, Clock's web page on product updates has a bullet point list that names the top changes in just a few words, and the "Read more" link takes the user to the web page with the full release notes ("Software update release notes," n.d.). Additionally, all of the companies included a table of contents with a comprehensive list of what the release note documentation contained. In PDFs, this appeared as a traditional table of contents at the beginning of the document. In web pages, the table of contents appeared as a left side navigation bar.

Another trend observed across companies is that release notes are housed in a section of the company website dedicated to product version information. This part of the website lists several recent versions of the same product with a link to the release notes for that version. Companies with multiple products might have all products and their versions listed in the same page or have version histories for different products on different pages. This location is significant because release note documentation is segregated from other parts of the company website, and a user navigates to the versions page specifically to review release notes. The format of listing several versions could make it difficult for a user who does not know which version of the software they are using, but it is helpful in quickly being able to navigate between versions if needed.

How Release Notes Are Framed

Inherent in publishing a collection of release notes, companies need to decide how to order the release notes and how to situate them in time. For ordering notes, including the biggest features or showiest enhancements first was seen across companies. Whether in web page or PDF form, the most attractive features were listed at the beginning, and later release notes were more often fixes, more technical, or had less self-evident benefits. For example, for the Blackboard Learn release notes published for 2018 Q4, the first release note is for a significantly re-designed user activity that changes how users interact with the system and introduces a new workflow, and the last release notes are for behind-the-scenes infrastructure changes ("Release notes for Blackboard Learn," n.d.). Also, four out of six companies included a list of additional changes at the end of the document. Instead of having full release notes, this was simply a brief list of other changes that were made in that software version and that did not require any setup or follow up. These appeared to be largely fixes and were not counted for this study as individual release notes.

The framing of a collection of release notes is also important in making sure that the user gets the correct note at the correct time. Because the purpose of software release notes are to upgrade to a specific software version, they have a strong temporal element that is unique for documentation. The companies studied addressed this need by titling the release note collections based on the name of the version, including the date of publication, and using other numeric sequencing. For example, the Clock release notes are displayed along a kind of vertical timeline so the user can scroll backward through the history of the software ("Software update release notes," n.d.).



Clock PMS Update

15 Oct 2018 Product updates

- WRS - Guest Profile Login
- Company - Custom Fields
- Charge Templates - Redisign and Custom Fields
- Billing To - Redesign
- Package elements – "First night only"
- GDPR - Guest Data Retention Period; "Forget" option
- Other improvements and Fixes

Read more ...

Clock PMS Update

30 Jul 2018 Product updates

- New Guest Profiles (GDPR compliant)
- Capacity counters - daily capacity
- Room Change and Housekeeping
- Company in Rate and Availability
- QuickBooks
- Saflok
- API - new endpoints
- Other improvements and Fixes

Read more ...

Clock PMS Update

31 May 2018 Product updates

- GDPR statements for personal data collection in the WRS, the Self Service and the Kiosk
- Kiosk - new setting regarding the mandatory Registration Card completion
- Default Folio email template for the Self Service Portal and Kiosk
- Room calendar colours for bookings
- Email list for locked user notifications
- Zapier
- Other improvements and Fixes

Read more ...

Figure 1. Release notes presented in a timeline format on the Clock website.

Chapter V: Results - Release Notes Individually

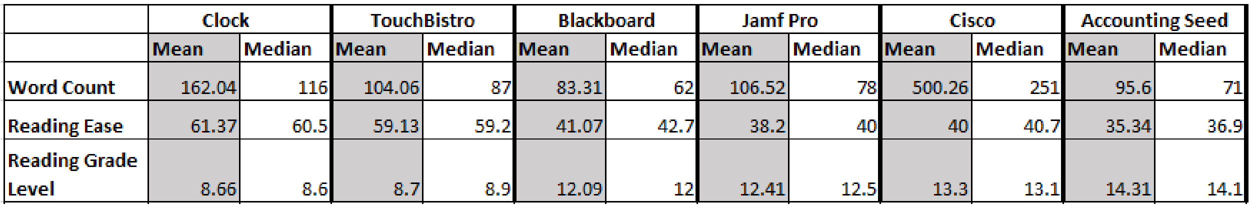
The second line of analysis seeks to understand more about individual characteristics of release notes. This examination dives into the details of release notes to show how their complexity, structure, visuals, and tone contribute to the genre.

Complexity

Both the linguistic and technical complexity of release notes in the corpus varied widely. The study by Hsia (2007) of technical airplane maintenance manuals ranked less complex manuals as higher quality because they reduced cognitive load for the reader. Hsia (2007) used six quality metrics to evaluate technical documentation readability: Sentence structure, sentence order, the order of semantic and syntactic forms, simple and complex sentences, examples, and graphics. For this study, linguistic complexity was gauged based on standard reading ease and grade level scales, and technical complexity of a document was observed based on its content and assumed level of knowledge.

Linguistic complexity. The Flesch Reading Ease scale uses the average sentence length and average number of syllables per word to determine a reading ease score on a 100-point scale, where a higher score indicates that the document is easier to read ("Test your document's readability," n.d.). The Flesch-Kincaid Grade Level scale uses the same metrics to calculate the U.S. school grade level of a document, where a lower score indicates a lower grade level ("Test your document's readability," n.d.).

The results of these measures for the release notes in the corpus are seen in Figure 2.



	Clock		TouchBistro		Blackboard		Jamf Pro		Cisco		Accounting Seed	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Word Count	162.04	116	104.06	87	83.31	62	106.52	78	500.26	251	95.6	71
Reading Ease	61.37	60.5	59.13	59.2	41.07	42.7	38.2	40	40	40.7	35.34	36.9
Reading Grade Level	8.66	8.6	8.7	8.9	12.09	12	12.41	12.5	13.3	13.1	14.31	14.1

Figure 2. Word count, reading ease, and grade level for release notes in the corpus.

Clock release notes have an eighth-grade reading level and have the lowest level of linguistic complexity with an average reading ease level of 61. The most linguistically complex of the release notes studied was Accounting Seed, with an average reading ease level of 35 and being at a college reading level. While this range might at first appear quite broad, the release notes are actually fairly closely clustered and point to an overall average of a high school reading level. It is interesting to note that Cisco and Accounting Seed are the most difficult to read and are also the two companies that publish release notes as PDFs. While the sample size is not large enough to draw a definitive correlation, this would suggest a possible relationship between the delivery format of the release notes and their ease of reading.

To support the communicative goal of successfully upgrading the software, release notes should be as understandable as possible. If release notes are challenging to understand due to high linguistic complexity, it is more difficult for readers to take appropriate actions based on the release note and could endanger the software upgrade. According to the U.S. Bureau of Labor Statistics, computer user support specialists is categorized as an occupation that requires less than a bachelor's degree for entry, but employers typically require some college (Fayer, Lacey, & Watson, 2017). Based on the Flesch-Kincaid Grade Level scale, the release notes analyzed are within an appropriate range of linguistic complexity for their audience.

Technical complexity. Beyond language, the other factor in the understandability of a release note is the technical complexity of the content itself. Software management is inherently a technically complex field, which is part of the reason why the release notes analyzed are written for IT analysts with specialized knowledge as opposed to a general audience. Also, the complexity of the release note content might be more reflective of the software product than a characteristic of the documentation. For example, analysts setting up TouchBistro can configure system options in an administrator-facing menu with a specially designed user interface ("What's new," n.d.), but Cisco analysts are expected to directly edit system setting code (Cisco Meeting Server release notes, n.d.).

The technical complexity also varies based on the level of assumed knowledge. The release notes reviewed showed that each company assumed that the reader of the release note has some level of familiarity with the software. This is evident in that the documentation does not begin with an explanation of what the software does or how it works overall, but the release notes rather go right into specific changes. It is appropriate for release notes to assume that the audience already knows generally what the software does and how it is used because the release notes are serving a specific purpose in helping organizations that are already using one version of the software to upgrade to a different version of the software. Manuals for first-time users or initial installation might need to include more explanatory and general contextual information to explain the overall goals and function of the software.

The second characteristic that reveals an expected level of software knowledge is by using software-specific terms without defining them. For example, the Jamf Pro release notes include names of specific features and functions, such as "Inventory Preload," "the Roster category," and "pre-stage enrollment," without explaining what they are or how they work ("Jamf Pro release notes," n.d.). Instead, the release notes assume that the audience is already familiar with these features of the software.

Release notes not only assume a level of knowledge about the specific software, but also a level of familiarity with technical concepts in general. In this way, the release notes in the corpus had a much broader spread for the amount of technical knowledge needed. The Clock, TouchBistro, and Accounting Seed notes would be fairly approachable for a general audience who is familiar with the software but doesn't have specialized technical training. The Blackboard release notes represent a middle level of knowledge by using standard technical terms that would be familiar to most analysts who have some experience in a previous IT role, such as concepts like cloud storage integration or dynamic/static files ("Release notes for Blackboard Learn," n.d.). The Jamf Pro release notes assume a much higher level of specialized technical knowledge. For example, they refer to cache size, APIs, and LDAP servers, which are general technical terms not specific to the product, but might be only known by an audience with technical experience or training ("Jamf Pro release notes," n.d.).

Move-Step Structure

Although there is wide range in release note length and complexity, structural patterns emerge by applying Swales' method of identifying rhetorical moves (1990). Figure 2 shows the variance in release note length based on word count. For example, one note from Clock is merely 18 words: "Company Credit Cards - the cards will now be kept until the next day after the card expiry date" ("Software update release notes," n.d.). In contrast, the first release note in Cisco version 2.5 is 1,408 words and included multiple subsections and tables to introduce complex new functionality for supporting multiple callers (Cisco Meeting Server release notes, n.d.).

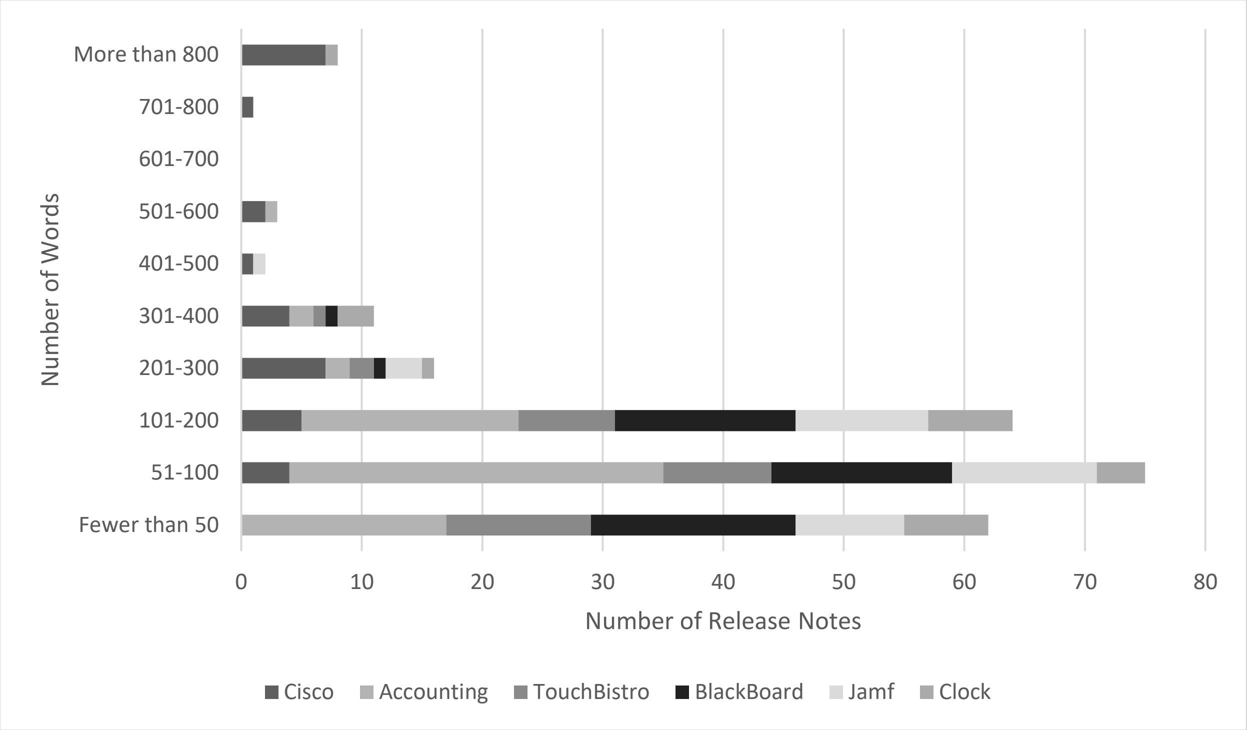


Figure 3. Range of word count for release notes in the corpus.

Despite the variance between the length and complexity of release notes, they follow a common pattern of moves to support their communicative purpose. Although not every release note contains every distinct move or step, the following common moves appear:

Move 1: Description

The opening move describes the change that is the focus of the release note, and it is often the bulk of the release note. The description can be comprised of several steps:

- Step: Background

This step places the change in the context of the user workflow or the location within the software. It orients the reader to how the change fits within the software.

- Step: Benefit

For new features and enhancements, this step can seem more promotional in nature, explaining why the change is beneficial to users and getting the reader excited about implementing the change. For fixes, this step might be implied, such as why the fixed behavior is better than the broken behavior, or entirely absent.

- Step: Previous/Now

This construct compares the previous behavior of the software to the changed behavior. It can have the background and benefit embedded within it or stand separately, and it serves the same communicative purpose of the move by explaining the change.

- Step: Example

This step contributes to the overall move of explaining the change by providing a specific example. By giving a scenario or use case, it illustrates an abstract change in a concrete and relatable way.

Move 2: Setup

One of the criterion for the release note corpus analyzed was that the notes had to necessitate user action, as opposed to informational only. Yet even with that condition, the software release notes published for a given version were typically a mix of change descriptions that required action and those that did not. Across the corpus, 49.4 percent of release notes included this move and required setup. Variance was low between the companies studied, and all six companies included setup steps in 42.4-56.7 percent of their release notes.

The purpose of this move is to tell the analyst who is responsible for the software upgrade how to act on the change described. For enhancements and new features, this could be how to turn on the feature or customize it for their organization. For fixes, this could be how to implement the fix so that the broken behavior is no longer experienced.

Move 3: Considerations

The communicative purpose of this move is to describe additional considerations that could affect a user's implementation of the change. This move was perhaps the most variable in presentation. It could be included at the beginning or end of the note. It might be embedded directly within the text of the release note or visually distinguished for emphasis, such as in a box or with an icon. The steps within this move are not all present within each release note, but all serve the same overall goal of this move.

- Step: Prerequisites

The considerations might include prerequisites to implementing the change, such as setup that must be completed first or other features that must be in use for the release note to be applicable.

- Step: Versioning

This steps calls out specific version information. Although release notes themselves are specific to a given version of the software, there might be additional version considerations, such as if the change is applicable only when upgrading from a certain version or only if your software is integrated with a second product that uses a particular version.

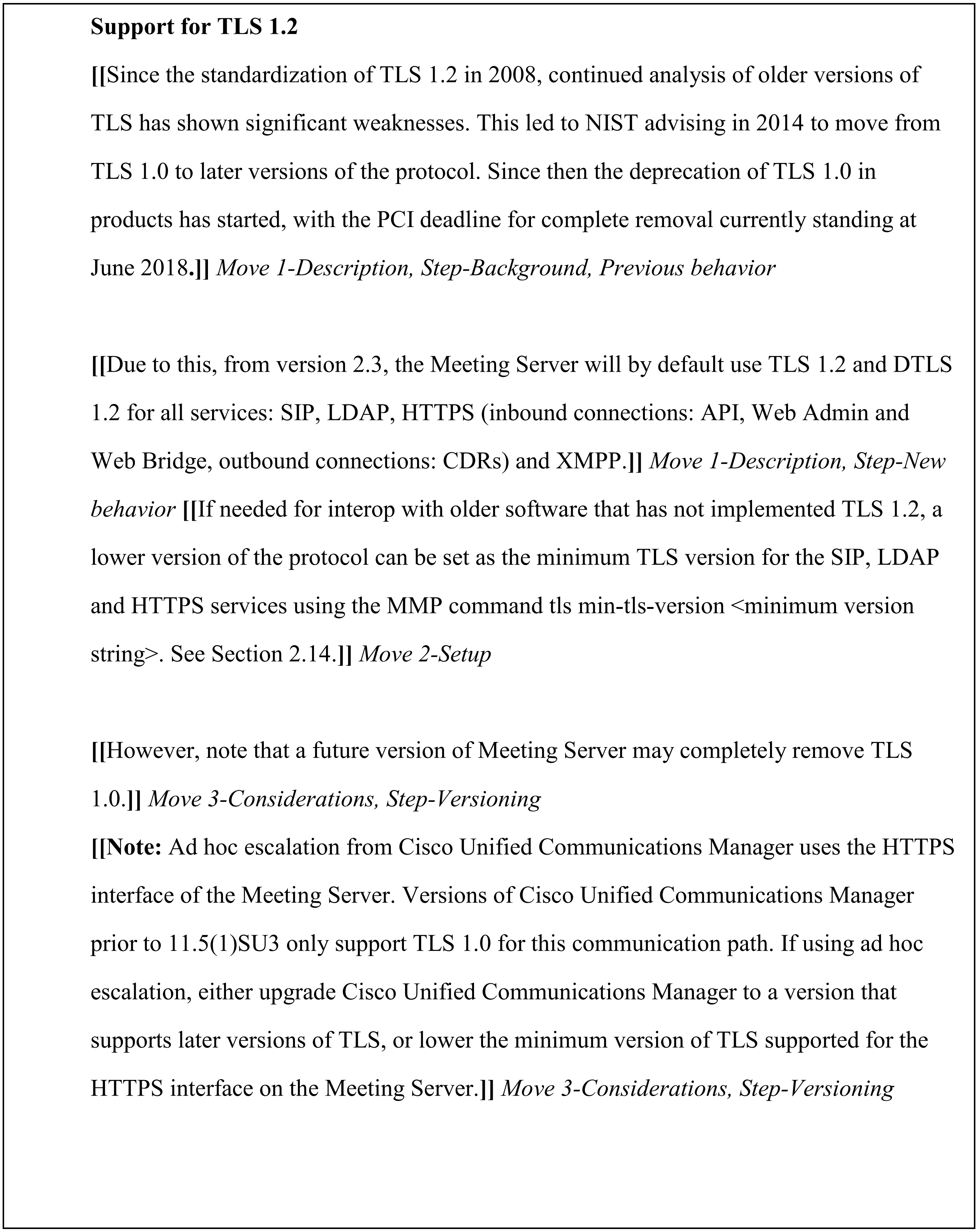
- Step: Limitations

This step describes the caveats of where the change might not apply or the constraints on what the change achieves. This step serves as a kind of counterpoint to Move 1, Step-Benefit by explaining when or why the benefit might not be fully realized.

- Step: Caution

If the change described presents a risk, this is highlighted in the considerations move. Examples of cautionary notes include situations where implementing the change could have unintended consequences or a certain step must be conducted carefully to prevent error. This step is most likely to have additional formatting, such as being in bold type or having an exclamation point icon next to it, to make sure that the reader does not miss this information.

Examples of move-steps in release notes. The following two examples illustrate the move-step structure often found in release notes. The first release note is from Cisco Meeting Server version 2.3, with moves and steps identified (Cisco Meeting Server release notes, n.d.):



Support for TLS 1.2

[[Since the standardization of TLS 1.2 in 2008, continued analysis of older versions of TLS has shown significant weaknesses. This led to NIST advising in 2014 to move from TLS 1.0 to later versions of the protocol. Since then the deprecation of TLS 1.0 in products has started, with the PCI deadline for complete removal currently standing at June 2018**.]]** *Move 1-Description, Step-Background, Previous behavior*

[[Due to this, from version 2.3, the Meeting Server will by default use TLS 1.2 and DTLS 1.2 for all services: SIP, LDAP, HTTPS (inbound connections: API, Web Admin and Web Bridge, outbound connections: CDRs) and XMPP.**]]** *Move 1-Description, Step-New behavior* **[[**If needed for interop with older software that has not implemented TLS 1.2, a lower version of the protocol can be set as the minimum TLS version for the SIP, LDAP and HTTPS services using the MMP command tls min-tls-version <minimum version string>. See Section 2.14.**]]** *Move 2-Setup*

[[However, note that a future version of Meeting Server may completely remove TLS 1.0.**]]** *Move 3-Considerations, Step-Versioning*

[[Note: Ad hoc escalation from Cisco Unified Communications Manager uses the HTTPS interface of the Meeting Server. Versions of Cisco Unified Communications Manager prior to 11.5(1)SU3 only support TLS 1.0 for this communication path. If using ad hoc escalation, either upgrade Cisco Unified Communications Manager to a version that supports later versions of TLS, or lower the minimum version of TLS supported for the HTTPS interface on the Meeting Server.**]]** *Move 3-Considerations, Step-Versioning*

Figure 4. Moves in Cisco release note.

This example from Cisco demonstrates the integration of multiple moves, while flowing in a predictable direction from a higher-level description to specific considerations.

The second example is a Jamf Pro version 10 release note with move and steps identified. While it includes the same basic moves as the Cisco release note analyzed above, it uses different steps to achieve the same communicative goals (“Jamf Pro release notes,” n.d.).

Email Inventory Reports for Advanced Computer Searches

[[You can now email inventory reports for advanced computer searches. Reports can be emailed immediately or according to a schedule.**]]** *Move 1-Description, Step-Benefit* **[[**You can configure multiple email reports to be sent to different recipients on different schedules.**]]** *Move 1-Description, Step-Example*

[[To access this feature in Jamf Pro, navigate to **Computers** > **Search Inventory** for a list of saved advanced computer searches, or create a new advanced search. Then, click the **Reports** tab.**]]** *Move 2-Setup*

[[Note: This enhancement replaces the export method for downloading saved advanced computer search reports.**]]** *Move 3-Considerations*
For more information, see the Computer Reports section in the *Jamf Pro Administrator's Guide*.

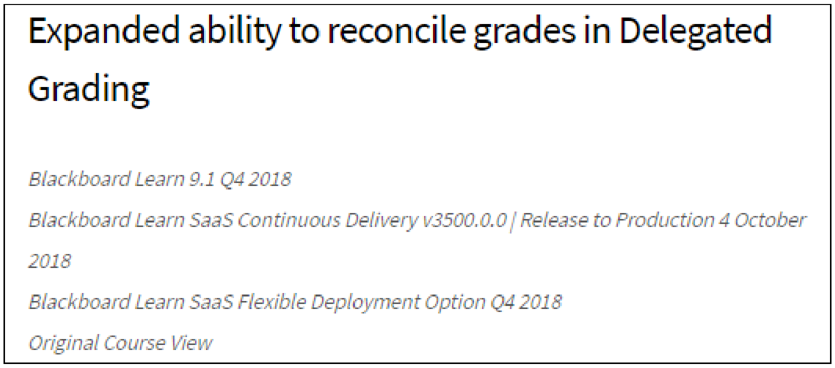
Figure 5. Moves in Jamf Pro release note.

Despite the many differences between the Cisco and Jamf Pro release notes, the same structural moves appear and serve the same communicative purposes. The overarching purpose unites release notes as a genre, and it is not surprising that they use similar structures to achieve their goal.

Other Structural Components of Individual Release Notes

Along with moves and steps, several other structural components were observed that illustrate both similarities and differences between software release notes. These elements include metadata, hyperlinks, and text formatting. Because software companies want their software to be upgraded correctly and because lower complexity is better when trying to support a reader in completing a task (Hsia, 2007), the companies have a vested interest in making sure that release notes are understandable to readers. To mitigate the inherent challenges of highly technical content, the release notes reviewed used these elements to improve understanding and help readers navigate the complexity.

Metadata. Before the release note begins, or between the title and the first paragraph, the release note might contain metadata. For example, Blackboard included relevant product names and version information in gray, italicized text underneath the release note title (“Release notes for Blackboard Learn,” n.d.):



Expanded ability to reconcile grades in Delegated Grading

Blackboard Learn 9.1 Q4 2018

Blackboard Learn SaaS Continuous Delivery v3500.0.0 | Release to Production 4 October 2018

Blackboard Learn SaaS Flexible Deployment Option Q4 2018

Original Course View

Figure 6. Versioning information included in a Blackboard release note.

The inclusion of metadata seemed to be a stylistic “all-or-nothing” choice by each company where it was either included for every release note or none of them.

One determining factor for whether a release note includes metadata might be the number of products that a certain company has and the number of versions available. Another factor might be the delivery mechanism and how the audience is expected to approach the release notes. If the release notes are published to a web page that a user might access from search results, metadata could be useful to help users know what product and version they are reading about, regardless of where they jump into the page. If printed as a PDF where all of the release notes are for the same product and version, this information is redundant and only needs to be stated once at the beginning of the document.

Formatting of setup instructions. For the release notes that included setup information, the instructions were typically presented in one of the following formats:

- Numbered steps. Complex setup instructions were presented sequentially, similarly to traditional instruction manuals, where a user completes step 1 before going to step 2, etc.
- Bulleted list. Setup instructions presented as bulleted lists were similar in purpose and complexity to numbered steps and the decision to use bullet points instead of a numbered list appeared to be more stylistic than a reflection of the content.
- Paragraph form. In instances where the setup consisted of a single step or a minimal number of steps, the setup might be written as a paragraph instead of separated as a list of steps.
- Links to setup instructions. Rather than including the setup instructions to implement a change within the release note itself, the release note might instead include a link to a user guide or setup manual for more information.

At times, these formats were used inconsistently, and in some cases, changing the format for the setup instructions might have helped improve clarity. Besides the complexity and length of the setup, other considerations when determining which format to use might be the amount of space on the page for PDF documents or consistency with the release notes before and after it.

Hyperlinks. As noted above, setup instructions might be included directly within a release note or be in a separate document that is linked from the release note. Blackboard did not include most setup instructions directly within the release note, but rather used the release note to describe the change, explain any limitations or additional considerations, and then provide a link to the setup instructions (“Release notes for Blackboard Learn,” n.d.).

Even for companies that did include setup instructions within the release note, links were often included to help the reader access more information about the product, such as a link to the main installation guide for the product. Links were used in both release notes presented in an online format and in those that were published as a PDF. For a new reader who is encountering these features or terminology for the first time, a link to a guide with more information can help them understand more context about the functionality discussed in the release note. The previous example of the Jamf Pro note included a link as the last line of the release note to take the user to additional information about the topic discussed in the release note. This information is not essential to understanding this change but could be helpful in providing supporting information for an analyst who is less familiar with the functionality.

Links allow release notes to be concise, while still providing users with additional information that they need just a click away. Instead of bloating release note documentation with details or related feature information not directly relevant to the current change, a link facilitates having that information in another place that is still easy for readers to find. However, a risk with a link is that a reader who clicks a link and goes into another document might not find their way back to the original release note and miss other important information. Also, a lot of links might make the documentation feel disjointed, and users might struggle to find the information that they are looking for after following a link.

Text formatting. Nearly all of the release notes reviewed used text formatting to call attention to particular information. Many of the release notes used bold typeface for screen names, button names, or system paths, or to emphasize certain information. However, an excessive use of bolding can undermine its effectiveness. A single release note published by Clock in July 2018 uses bold typeface to mean three different things ("Software update release notes," n.d.):

- The first sentence bolds the key words, likely to help a skimming reader catch the benefit of the change: "Our new Guest Profile system enables you to **define any kind of fields** to be contained in Guest Profiles, Registration Cards and the Register."
- The release note next includes a bulleted list where the first words of each item on the list are in bold, such as:
    - "**Date.** The field is for dates checking their format."
    - "**Phone**. The field is for phone numbers checking their format."
- Then, the release note uses bold for a system path and a step that a user needs to complete: "To set up these fields, go to **'Settings' - 'Guest Profile Fields'**."

In comparison to bold typeface, italics was used much more sparingly and no text underlining was observed.

Not only were bullet points used in setup instructions, but also in descriptions and other parts of the release note. Bullet points have been recognized as a design feature to improve the readability of complex texts (Raynor & Dickinson, 2009). In the example above, Clock used a bullet point list for the various fields supported ("Software update release notes," n.d.). Other release notes used bulleted lists for multiple aspects of a change, multiple benefits, or other similar items.

Headings. A common feature in release notes was the use of headings or subheadings to break notes into sections. Whether subheadings were used within a release note seemed to be determined based on the length and complexity of the release note. Short release notes had only a title followed by a description. Longer notes might have a subheading after the description to begin the setup steps.

Very long release notes, such as those over several hundred words and introducing a major change, might have multiple subsections within the release note to describe different aspects of the change and the setup needed for that specific aspect. The lengthy Cisco release notes provide an example of using subheadings to divide a single release note into smaller sections. In version 2.4, release note 2.3-Pane Placement includes several subsections, each demarcated with a numbered subheading in smaller font (Cisco Meeting Server release notes, n.d.):

- 2.3.1-Similarities and differences between assigning importance and pane placement
- 2.3.2-Using pane placement
- 2.3.3-Removing pane placement
- 2.3.4-Example of using pane placement

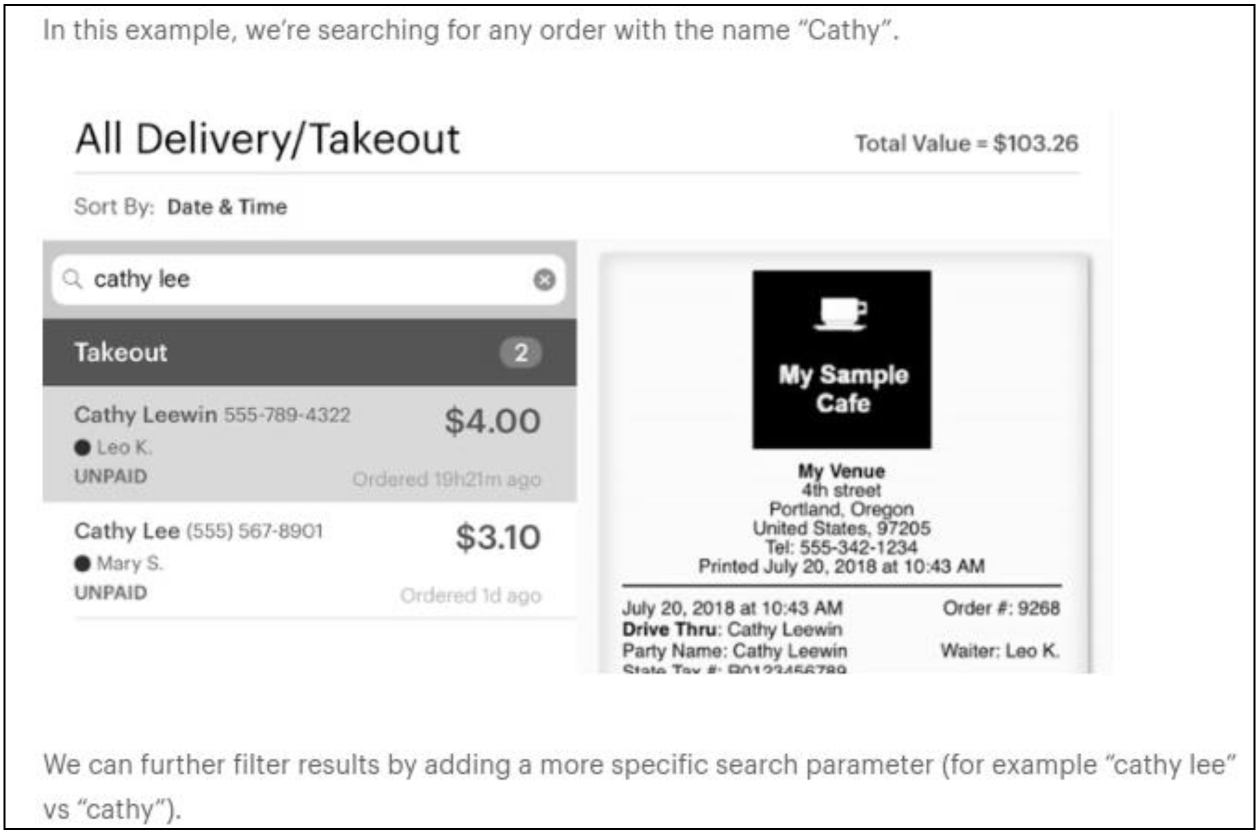
Because the entire release note is six pages in length, these subsections contribute to the communicative goals of the release note by helping the reader parse the different aspects of the pane placement change and navigate to the information they need.

Visuals in Release Notes

The release notes in the corpus used different kinds of visual elements and to varying extents. The three main types of visuals observed in individual release notes were screenshots, diagrams, and high importance callouts. TouchBistro included at least one screenshot in every release note (“What’s new,” n.d.). Accounting Seed and Jamf Pro represented the other end of the spectrum with zero visuals, and the other companies studied were in between.

Screenshots. A screenshot is an image of how the software interface appears. The screenshot could be of an administrator activity that would be accessed only by analysts, or it could be of an end-user facing activity.

TouchBistro relied on screenshots as a core part of understanding the change in the release note (“What’s new,” n.d.). The screenshots were accompanied by text, but a reader would struggle to understand the description without the screenshot. In the following example from TouchBistro version 8.1, the text does not stand independently, and the screenshot is necessary to understand the example (“What’s new,” n.d.):



In this example, we're searching for any order with the name "Cathy".

All Delivery/Takeout

Total Value = $103.26

Sort By: Date & Time

cathy lee

Takeout 2

Cathy Leewin 555-789-4322 $4.00
● Leo K.
UNPAID Ordered 19h21m ago

Cathy Lee (555) 567-8901 $3.10
● Mary S.
UNPAID Ordered 1d ago

My Sample Cafe

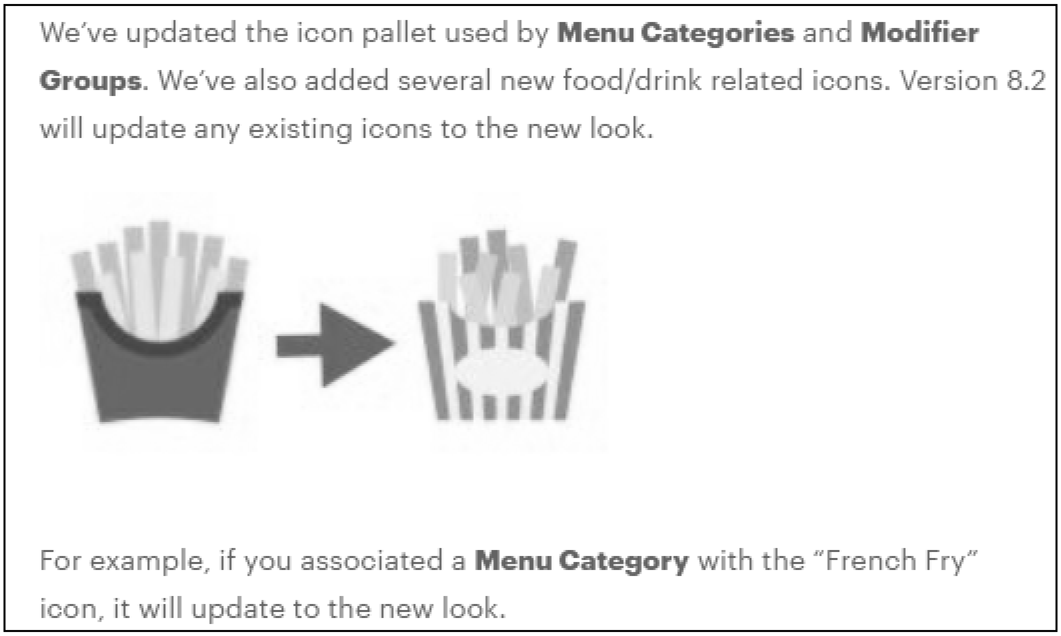
My Venue
4th street
Portland, Oregon
United States, 97205
Tel: 555-342-1234
Printed July 20, 2018 at 10:43 AM

July 20, 2018 at 10:43 AM Order #: 9268
Drive Thru: Cathy Leewin
Party Name: Cathy Leewin Waiter: Leo K.
State Tax #: R0123456789

We can further filter results by adding a more specific search parameter (for example "cathy lee" vs "cathy").

Figure 7. Screenshot from TouchBistro 8.1 release note.

TouchBistro also used screenshots to show before and after images of the changes, especially in cases where it would be difficult to explain the change using words alone ("What's new," n.d.). These before and after images help orient the user to the old look and contrast it to the new look. In this example from TouchBistro version 8.2, having both the old icon and the new icon illustrates the style of the aesthetic change but does not provide an exhaustive list of all the updated icons ("What's new," n.d.).

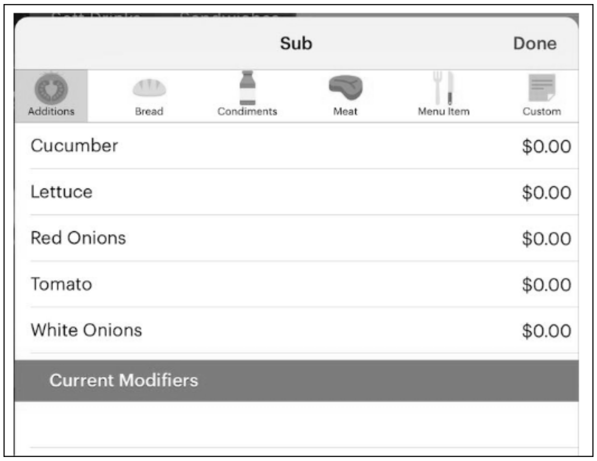


We've updated the icon pallet used by **Menu Categories** and **Modifier Groups**. We've also added several new food/drink related icons. Version 8.2 will update any existing icons to the new look.

For example, if you associated a **Menu Category** with the "French Fry" icon, it will update to the new look.

Figure 8. Before and after image in a TouchBistro release note.

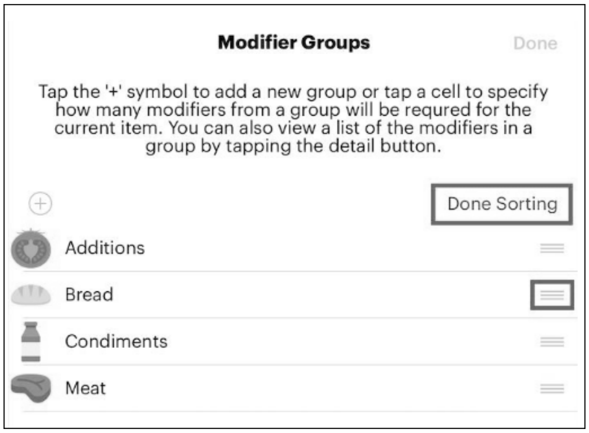
TouchBistro also used both native and non-native annotations to call attention to certain screen elements ("What's new," n.d.). This tactic helps the reader to know where to look in the image, especially if the image is illustrating specific steps. In Figure 7, the screenshot uses highlighting that appears in the system to show which item to select. In Figure 8, the screenshot is annotated with red boxes to show users the next option to tap.



Cucumber	$0.00
Lettuce	$0.00
Red Onions	$0.00
Tomato	$0.00
White Onions	$0.00

Current Modifiers

Figure 9. On-screen highlighting.



Modifier Groups

Tap the '+' symbol to add a new group or tap a cell to specify how many modifiers from a group will be requred for the current item. You can also view a list of the modifiers in a group by tapping the detail button.

Figure 10. Red boxes show where to tap.

These examples are drawn from TouchBistro because they were the most prolific in the use of screenshots in the release notes studied. The various ways that TouchBistro used screenshots, including as examples, before and after images, and with annotations, was also observed in release notes from Cisco, Blackboard, and Clock.

Diagrams. Cisco used diagrams to depict complicated setup scenarios. These release notes included full descriptions of the behavior and used a supplementary diagram to help clarify or illustrate an abstract concept. An example of this is in a Cisco Meeting Server version 2.4 release note that describes two options for load balancing calls and includes a diagram to illustrate (Cisco Meeting Server release notes, n.d.).

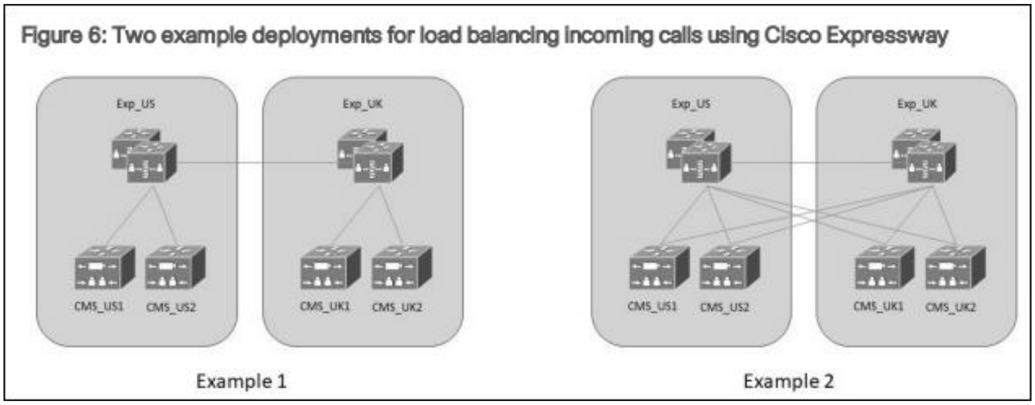


Figure 6: Two example deployments for load balancing incoming calls using Cisco Expressway

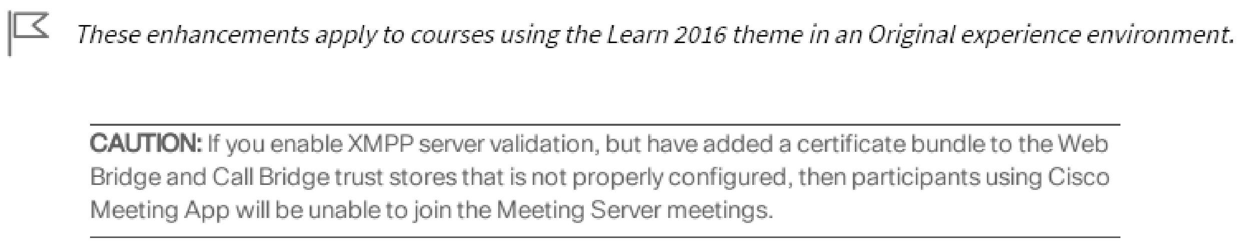
Figure 11. Diagram in a Cisco release note to show two setup options.

These Cisco diagrams differ from screenshots in a few ways. Most obviously, they are created outside of the software to depict a concept, instead of capturing the visual appearance of the actual software. Although the diagrams might aid decision making or setup strategy at a high level, they do not provide a visual aid to specific setup steps. The diagrams also give the authors of the release notes more control over how to label, annotate, or highlight certain aspects, without the background noise of the rest of the software screen. Also, Cisco used diagrams alongside the text, but the text could also stand alone. In contrast, TouchBistro screenshots were essential to understanding. Despite these differences, both screenshots and diagrams served the same overarching purpose of aiding a reader in understanding the change described in the release note so that they could take the appropriate next step.

The Cisco release notes certainly used the most diagrams. This volume could be merely a stylistic preference, but because the Cisco release notes are much more technically complex, there seems to be a much greater need for diagrams (Cisco Meeting Server release notes, n.d.). As seen in the examples included above, the TouchBistro screenshots showed users what to click or tap in the software itself, but the Cisco diagram was to aid with higher-level decision making, and this pattern was seen throughout the corpus.

Caution indicators. The third move identified in the move-step analysis above was "notes" with version information, considerations, or other limitations. In four of the six release notes studied, these "notes" were preceded with the word Note in bold text, followed by the note in normal text. Accounting Seed instead used the word Note and italicized the entire statement ("Release notes," n.d.), and this move was not visually called Note in Clock release notes ("Software update release notes," n.d.).

Blackboard and Cisco also differentiated between typical and high-impact "notes" by using a different visual indicator for the most important callouts, instead of just labeling them "notes." Figure 10 below shows how Blackboard called attention to these using a flag icon ("Release notes for Blackboard Learn," n.d.), and Cisco placed these in a box and used red, bold text to label them as "Caution" (Cisco Meeting Server release notes, n.d.).



These enhancements apply to courses using the Learn 2016 theme in an Original experience environment.

CAUTION: If you enable XMPP server validation, but have added a certificate bundle to the Web Bridge and Call Bridge trust stores that is not properly configured, then participants using Cisco Meeting App will be unable to join the Meeting Server meetings.

Figure 12. Cautionary notes. Top image: A “note” marked by a flag in a Blackboard release note. Bottom image: A caution “note” with special formatting in a Cisco release note.

While the purpose of the screenshots and diagrams described above were to assist understanding, these visual elements for cautionary notes serve a different role to make sure that the user does not miss this information. They follow some of the best practices in caution technical writing through a signal word panel design that is set apart from the rest of the text, and has something on the left side to catch the reader’s eye (Roe, 2010). The Blackboard note also includes a flag, although it is not obvious that the icon merits additional attention, and the Cisco note uses “Caution” in red text as a signal word (Roe, 2010). These attributes make the warnings easy to see while skimming and are intended to stand out to all users. Release notes from the other companies studied did not have an equivalent visual indicator for the most important information. That could be appropriate if there is no information that needed such emphasis in the notes reviewed, or it could be a design flaw if there is vital information that is buried within the text of the release note and a missed opportunity for making that information stand out.

Tone of Individual Release Notes

The release notes analyzed achieved different tones by how they referred to themselves and their audience and other word choices.

Actors. Software release notes that meet the criteria for this study have multiple agents, either stated or implied. The different ways that software companies referred to these agents in the release notes shows how they relate to the audience and affects the tone of the document. Figure 11 below summarizes the terminology that each company used for each of the agents. Note that terminology is used inconsistently even within the release notes produced by the same company, and Figure 11 shows the most often used terms.

	Software Company	**Software Product**	**Customer Organization**	**Reader of the Release Note**	**End User at Organization**
Accounting Seed	Accounting Seed support, We	Accounting Seed	You	You, Existing users	Users
Blackboard	Blackboard	Blackboard, Blackboard Learn	Institution	Administrators	Users, Instructors, Course designers, Students
Cisco	We	Meeting Server	Company	You	Participant, User
Clock	We	N/A	You	You	User
Jamf Pro	Jamf Pro	Jamf Pro <version>	You	You	User
TouchBistro	We, TouchBistro	TouchBistro	You	You	You

Figure 13. How release notes by each company referred to the roles involved.

Based on this analysis, the most common way for these companies to refer to the reader is “you.” For example, a release note in Jamf Pro version 10.10 states, “You can now add configuration profiles to a computer PreStage enrollment” (“Jamf Pro release notes,” n.d.). This use of the second person directly addresses the reader and makes them responsible for the action. It might also suggest a more personal relationship.

Blackboard is the exception to using “you” and instead referred to the reader as administrators (“Release notes for Blackboard Learn,” n.d.). Blackboard also stands out because it used the most impersonal language to speak of itself as “Blackboard” instead of the more personal “we” that other companies used (“Release notes for Blackboard Learn,” n.d.). For example, a release note from Blackboard 9.1 Q2 2018 states, “Blackboard is committed to protecting the privacy of all users" ("Release notes for Blackboard Learn," n.d.). Blackboard also differs from the other release notes by referring to users by more specific roles. Instead of referring generically to "users" as other companies did, Blackboard distinguishes whether the change affects instructors, course designers, or students ("Release notes for Blackboard Learn," n.d.). This differentiation might reflect the specific kind of software that Blackboard is producing and the separation between types of users, and it helps explain who the change affects.

Accounting Seed was one of the organizations that referred to software end users generally as "users" ("Release notes," n.d.). However, the Accounting Seed release notes refer to the organization using the software and the reader of the release note as "existing Accounting Seed users," "existing users," or "you" ("Release notes," n.d.). For example, an Accounting Seed Echinacea release note states: "Existing users should add the Comment field to your Recurring Billing Line page layout." ("Release notes," n.d.). This administrative task would be completed by the analyst responsible for reading the release note and upgrading the software, but the generic term "existing users" could be read as this task being for the end user of the software instead, even though they are not the audience for the release note.

All of the release notes reviewed used a significant amount of passive voice and often did not identify themselves or the software as taking the action. The Accounting Seed release notes for the Clover version exemplify this as 16 out of 24 release notes used passive voice. For example, the first sentence of one release note states: "A new field has been added on the Accounting Settings Page to capture a default Vendor account" ("Release notes," n.d.). In contrast, a Clock release note described a similar change but used active voice: "We have added a new setting for the Kiosk to require passport pictures or not" ("Software update release notes," n.d.). The more active use of "we" connects the software company to the user directly, instead of the disengagement of passive voice.

Yet the release notes were inconsistent within a given company, version, and even within the same release note, such as changing between first and third person or switching into and out of passive voice. This inconsistency could suggest that how the company addresses actors might not be a conscious choice. Or, it could indicate that the software company relates differently to the audience depending on the type of change, such as using the impersonal third-person or passive voice to distance themselves from fixes, and using the personal first-person to connect with the audience for popular enhancements. For example, a release note for TouchBistro includes the sentence "The Revenue Reports section has been removed" but later in the release note it says, "We've added a new Menu section with reports focusing on menu-item related analytics" ("What's new," n.d.). Perhaps TouchBistro used passive voice to distance themselves from the change to remove functionality, which users might not like, and used active voice to claim credit for the new functionality that might be more favorable among users.

Word choice. The vocabulary and sentence structures used created different tones between the release notes, such as more formal Cisco release notes and more casual and friendly Clock release notes. This difference in tone can be seen by comparing excerpts from release notes by each company:

> Cisco Meeting Server version 2.4.0:
>
> The Uploader component simplifies the work flow for uploading Meeting Server recordings to the video content manager, Vbrick, from a configured NFS connected to a Meeting Server. No manual importing of recordings is required.

> (Cisco Meeting Server release notes, n.d.)

> Clock PMS, Oct. 15, 2018:
>
> Until now, in every rate or package element, you had to enter all the necessary data like tax, revenue category, etc. This has changed and instead of repeated entry of one and the same information, now you can simply choose a charge template.
>
> (“Software update release notes,” n.d.)

Both of these release notes are describing new functionality to streamline existing workflows but achieve different tones through their word choice. The Cisco release note describes the change from the perspective of the system to discuss how recordings are uploaded and has a more formal and professional tone. It uses more technical system terms and advanced vocabulary, such as “configured NFS” and “manual import” (Cisco Meeting Server release notes, n.d.). The Clock release note has a more casual and approachable tone by describing the change from the perspective of the user and their workflow (“Software update release notes,” n.d.). It also uses the abbreviation “etc.” and the colloquial phrase “one and the same,” which are less formal language (“Software update release notes,” n.d.). Other Clock release notes have similarly casual language, using contractions, simplified terms, and more conversational phrasing (“Software update release notes,” n.d.). Given this tone, it is not surprising that Clock release notes were also found to have the lowest reading level based on the Flesch-Kincaid Grade Scale, as discussed in the previous Complexity section.

The directness of the release note also affected the tone. Clock release notes were very direct about why they made each change. All companies included some type of benefit for changes, but Clock explicitly stated it as the reason for the change. A common phrase in Clock release notes is “our goal,” such as in an October 2018 release note that stated: “Our goal is to make the connection between invoices and companies clearer” (“Software update release notes,” n.d.). By preceding the benefit statement with “our goal,” Clock is transparent about pulling down the curtain to explain why they made a change, which further personalizes the communication (“Software update release notes,” n.d.).

Another element that affected the tone of release notes was their use of examples. The TouchBistro release notes illustrated changes not only with screenshots but also with examples and scenarios where they named fictional people involved. A release note for TouchBistro version 8.2 described a change using the example of Leo K., the bartender (“What’s new,” n.d.). By including specific and relatable examples, the release notes also implicitly conveyed the company’s understanding of their audience and put the software company on the same side as the users. Naming people in their examples and including specific details also helped humanize the change and connect the abstract change to a concrete and relatable scenario.

An interesting trend appears in how the software companies studied use “please” in their release notes. Jamf Pro did not use “please” at all in the release notes that were part of this corpus (“Jamf Pro release notes,” n.d.). The other five companies used it one to four times in each collection of release notes, but nearly every time was in the context of where to seek additional information. For example, a Blackboard release note says “For more information, please see behind the Blackboard” (“Release notes for Blackboard Learn,” n.d.), and Accounting Seed includes “Please contact Accounting Seed support” for a specific change (“Release notes,” n.d.). The exceptions to this were cases where readers were told to “please note” a specific consideration or limitation. Using “please” when directing the reader to another resource or to reach out for support might be a way to make a more personal invitation. Not using “please” in other context, such as to “please” complete these steps or implement the feature, suggests the overall more professional nature of the documentation and why the changes and actions are mandatory for the upgrade.

Chapter VI: Discussion, Recommendation, and Conclusion

This genre analysis of software release notes has examined release notes as a genre both by their textual features and by their context and purpose, which Chandler (1997) suggested is inherent in the very definition of a genre. As Chandler states in his introduction to genre theory, "In addition to textual features, different genres also involve different purposes, pleasures, audiences, modes of involvement, styles of interpretation and text-reader relationships" (1997). By evaluating release notes through these lenses, this study has uncovered similarities in how release notes are written and unique attributes compared to other forms of documentation so that the research questions can be answered. This section summarizes those findings and suggests directions for further research.

RQ1 - What Similarities Unite Software Release Notes As a Subgenre?

The six companies studied represented a range of industries, yet the release notes are united in several ways. The release notes shared a common purpose to help existing customers of the software upgrade to a new software version, which resulted in the following similarities:

- Release notes were framed as fixes or enhancements, based on familiarity with previous system behavior.
- Release notes were intended to be read by analysts and did not include end-user instructions.
- Collectively, release notes were typically presented with a brief summary before the full release notes, the most impressive enhancements listed first, and a table of contents to ease navigation.
- Individually, release notes were typically structured with some combination of Move 1: Description, Move 2: Setup, and Move 3: Considerations.

- Release notes used common techniques to clarify complex information, including subheadings, bullet points, and links.

Because all of the release notes were trying to help similar users accomplish a similar purpose, it is not surprising that they use similar structures and formatting features to do it.

Yet there were also significant differences between the release notes studied, stemming both from product needs and company-specific style choices. Perhaps the dominant difference among release notes was their varying levels of complexity and length. Some release notes were much more technically complex, assuming a higher level of familiarity not only with the software product but also with general technical concepts. This level of complexity might be necessary to successfully upgrade the software and assumes that a reader in that role is capable of doing it. Other differences reflect stylistic choices by software companies, such as including metadata or special text formatting. The use of visuals varied widely, and a genre convention does not appear to have yet emerged for when or how to use screenshots or other visuals with textual information. Tone and word choice were also varied across the release notes sampled, perhaps reflecting the technical complexity of the product or the overall company culture, such as a more playful tone.

RQ2 - What Attributes Are Unique to Release Notes As Part of Accomplishing Their Communicative Purpose?

Both in the words of the companies producing release notes and from the organizations employing staff to process release notes, the release notes serve a clear communicative purpose to facilitate software upgrades. This goal leads to some of the same best practices that are used by traditional instruction manuals, such as warning boxes, numbered steps, and other text formatting to highlight key information.

Yet there are also a few characteristics that are unique to software release notes and are unlikely to be found in other genres:

- Release notes assume that the reader is already familiar with the software system. Because release notes are targeted for upgrades, they can assume that the user is not new to the system and doesn't require comprehensive information, but only needs to be informed of version-specific changes or setup.
- Release notes are written for a technical audience instead of the end user of the software.
- Certain types of information provided are exclusive to this genre, such as how to reverse an automatic change ("kill switch") so that analysts can back out a software change before it automatically takes effect and including beta features or steps for previewing a feature before full development is complete.

These attributes contribute to the overall communicative goal of release notes, but they also differentiate release notes from traditional installation and user manuals.

Contribution to the Field

This study demonstrates the flexibility of Bhatia's model for genre analysis and contributes to the large number of studies that have demonstrated its usefulness and applicability in a variety of contexts, such as seen by Singh, Shamsudin, and Zaid (2012).

For software companies and technical writers, this study contributes new insights into how release notes function as a genre. Given the prevalence and criticality of software release notes for facilitating software upgrades, this study is applicable across industries to lay a foundation for understanding release notes. Previous studies have failed to evaluate software release notes as a form of technical writing, focusing rather on developer change logs or utilities for release note production (Abebe, Nasir, and Hassan, 2016; Moreno, et al, 2016). With a better understanding of the release note subgenre, technical writers can begin developing best practices and guidelines for more effective release notes by considering their unique purpose and context. Instead of every company developing their own release note style, there is an opportunity to learn from others in the field and proliferate release note characteristics that are appropriate for the genre.

Software companies that are looking for ways to improve communication with organizations for smoother upgrades (Jokela & Jantti, 2012) can use the findings in this study to evaluate their own release notes. They might want to consider how their release notes account for the unique point in time, the audience that they are addressing, and how that audience is different from end users. They might also find characteristics such as the ordering of release notes and using visuals to be helpful improvements for their release notes.

Software continues to expand its role throughout industries and organizations, necessitating more software upgrade, and more release notes. As of May 2018, 42 percent of technical writers were employed in the computer systems design and related services industry, where job responsibilities likely include software documentation (Bureau of Labor Statistics, 2019). The findings in this study can provide a basis for training new technical writers about the fundamentals of release note writing so that they are prepared for future employment with software companies and equipped to deliver the information that users need.

Recommendations

Release notes also present many opportunities for further research. Future studies could perform usability testing to determine release note elements that are the most effective for users, such as specific structural or formatting strategies that improve user understanding. It would also be interesting to look at release notes longitudinally through the lifecycle of a product to track how the release notes change over time and as the software product matures. There is also a wide world to explore beyond software system release notes, which were the focus of this study, and look at release notes in other contexts, such as for phone applications or video games. Any of these directions could build on the groundwork established in this study to understand the purpose of release notes and their attributes that distinguish them from traditional instruction manuals or other forms of technical communication.

Conclusion

Software release notes play a significant role in how organizations upgrade their software, and the impact of release notes is seen throughout many industries. To correct the dearth of literature about software release notes thus far, this genre analysis examines similarities between release notes and unique characteristics that separate them from other genres. The findings show several consistencies that transcend a myriad of differences in pursuit of the same communicative purpose. Because software release notes are still a relatively young subgenre, more conventions might emerge as the genre matures. This study also demonstrates how release notes are distinct from other forms of technical documentation and require a different approach from practitioners and educators.

References

Abebe, S. L., Nasir, A., & Hassan, A. E. (2016). An empirical study of software release notes. *Empirical Software Engineering, 21*(3), 1107-1142. doi:10.1007/s10664-015-9377-5

Accounting Seed Knowledge Base. (n.d.). *Release notes.* Retrieved from https://support.accountingseed.com/hc/en-us/articles/115013879487-Release-Notes

Artemeva, N. (2005). A time to speak, a time to act: A rhetorical genre analysis of a novice engineer's calculated risk taking. *Journal of Business and Technical Communication, 19*(4), 389-421. https://doi.org/10.1177/1050651905278309

Barth, A. (2017). High-tech economy continues to grow with need for software skills. *The State Journal*. Retrieved from https://login.ezproxy.lib.uwstout.edu/login?url=https://search-proquest-com.ezproxy.lib.uwstout.edu/docview/1962166496?accountid=9255

Bhatia, V. K. (1993). *Analysing genre: Language use in professional settings.* London: Longman.

Bhatia, V. K. (2008). Genre analysis, ESP and professional practice. *English for Specific Purposes, 27*, 161-174. doi:10.1016/j.esp.2007.07.005

Blackboard Help. (n.d.). *Release notes for Blackboard Learn.* Retrieved from https://help.blackboard.com/Learn/Administrator/Hosting/Release_Notes/Release_Notes_All_91_Releases/

Bookcase (5926) – Assembly and Operation Manual. (2015). *MDB Family*. Retrieved from https://secure.img1-fg.wfcdn.com/docresources/472/23/234673.pdf

Bureau of Labor Statistics. (2019). 27-3042 Technical writers. *Occupational Employment and Wages, May 2018.* Retrieved from www.bls.gov/oes/current/oes273042.htm

Buse, R., & Weimer, W. (2010). Automatically documenting program changes. Proceedings from ASE ‘10: *IEEE/ACM International Conference on Automated Software Engineering.* Antwerp, Belgium: New York. doi: 10.1145/1858996.1859005

Chandler, D. (1997). *An introduction to genre theory*. Retrieved from https://www.researchgate.net/publication/242253420_An_Introduction_to_Genre_Theory

Cisco Meeting Server release notes. (n.d.). *Cisco product support.* Retrieved from www.cisco.com/c/en/us/support/conferencing/meeting-server/products-release-notes-list.html

Cisco Meeting Server with Cisco Expressway deployment guide. (2016). *Cisco product support*. Retrieved from https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/expressway/config_guide/X8-11/Cisco-Meeting-Server-2-4-with-Cisco-Expressway-Deployment-Guide_X8-11-4.pdf

Clock. (n.d.) *Software update release notes for Clock PMS and Clock POS.* Retrieved from www.clock-software.com/blog/software-updates/

Coney, M. B. (1992). Technical readers and their rhetorical roles. *IEEE Transactions on Professional Communication, 35*(2), 58-63. doi: 10.1109/47.144864

Department of Defense. (2010). Coverings for waterborne main propulsion shafting on U.S. naval surface ships and submarines. Retrieved from https://www.dcma.mil/Portals/31/Documents/NPP/MIL-STD-2199A.pdf

Dekel, U., & Herbsleb, J. D. (2009). Improving API documentation usability with knowledge pushing. *Software Engineering, ICSE 2009*. doi: 10.1109/ICSE.2009.5070532

Ding, H. (2007). Genre analysis of personal statements: Analysis of moves in application essays to medical and dental schools. *English for Specific Purposes, 26*. 368-392. doi:10.1016/j.esp.2006.09.004

Ding, W., Liang, P., Tang, A., & Van Vilet, H. (2014). Knowledge-based approaches in software documentation: A systematic literature review. *Information and Science Technology, 56*(6), 545-567. https://doi.org/10.1016/j.infsof.2014.01.008

Duff, D. (Ed.). (2014). *Modern genre theory.* Abingdon: Routledge.

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: Past, present, and future. *U.S. Bureau of Labor Statistics.* Retrieved from www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf

Flowerdew, J., & Wan, A. (2006). Genre analysis of tax computation letters: How and why tax accountants write the way they do. *English for Specific Purposes, 25,* 133-153. doi:10.1016/j.esp.2005.03.005

Frutos, A. C. (2015). A corpus-based genre study of instruction manuals for household appliances. *Procedia - Social and Behavioral Sciences, 198*, 103-111. doi: 10.1016/j.sbspro.2015.07.425

Grade assignments and provide feedback. (n.d.) *Blackboard help*. Retrieved from https://help.blackboard.com/Blackboard_Open_LMS/Teacher/Content/Activities/Assignments/Grade_Assignments_and_Provide_Feedback

Hsia, T. (2007). Evaluating the writing quality of aircraft maintenance technical orders using a quality performance matrix. *International Journal of Industrial Ergonomics, 37*(7), 605-613. https://doi.org/10.1016/j.ergon.2007.03.008

Hyon, S. (2017). *Introducing genre and English for specific purposes.* London: Routledge.

KitchenAid. (2009). *Instructions and recipes for your KitchenAid stand mixer.* Retrieved from https://www.kitchenaid.com/content/dam/global/documents/201705/UseandCareGuide-9709785-RevE.pdf

Jamf Pro. (2011). *Preparing to upgrade Jamf Pro.* Retrieved from https://www.jamf.com/jamf-nation/articles/136/preparing-to-upgrade-jamf-pro

Jamf Pro. (n.d.). *Jamf Pro release notes.* Retrieved from www.jamf.com/resources/product-documentation/jamf-pro-release-notes/

Jokela, K., & Jantti, M. (2012). Challenges and problems in product portfolio release and deployment management: A case study. Proceedings from *2012 9th International Conference on Service Systems and Service Management (ICSSSM).* doi: 10.1109/ICSSSM.2012.6252208

Klepper, S., Krusche, S., & Bruegge, B. (2016). Semi-automatic generation of audience-specific release notes. *2016 International Workshop on Continuous Software Evolution and Delivery*. doi: 10.1145/2896941.2896953

Luzon, M. J. (2005). Genre analysis in technical communication. *IEEE Transactions on Professional Communication, 48*(3), 285-295. doi: 10.1109/TPC.2005.853937

Microsoft Office. (n.d.). *Test your document's readability.* Retrieved from https://support.office.com/en-ie/article/test-your-document-s-readability-85b4969e-e80a-4777-8dd3-f7fc3c8b3fd2

Miller, C. (1984). Genre as social action. *Quarterly Journal of Speech, 70*(2), 151-167. https://doi.org/10.1080/00335638409383686

Monster Jobs. (n.d.) *Monster jobs.* Retrieved from www.monster.com/jobs

Moreno, L., Bavota, G., Di Penta, M., Oliveto, R. Marcus, A., & Canfora G. (2016). ARENA: An approach for the automated generation of release notes. *IEEE Transactions on Software Engineering, 43*(2), 106-127. doi: 10.1109/TSE.2016.2591536

Morgan, H. M., & Ngwenyama, O. (2015). Real options, learning cost and timing software upgrades: Towards an integrative model for enterprise software upgrade decision analysis. *Int. J. Production Economics, 168*, 211-223. http://dx.doi.org/10.1016/j.ijpe.2015.06.028

Naval propulsion program (NPP) training. (n.d.). *Defense Contract Management Agency.* Retrieved from https://www.dcma.mil/NPP/training/

Parkinson, J. (2017). The student laboratory report genre: A genre analysis. *English for Specific Purposes, 45,* 1-13. https://doi-org.ezproxy.lib.uwstout.edu/10.1016/j.esp.2016.08.001

Phoha, V. (1997). A standard for software documentation. *Computer*, *30*(10), 97-98. doi: 10.1109/2.625327

Raynor, D. K., & Dickinson, D. (2009). Key principle to guide development of consumer medicine information – Content analysis of information design texts. *The Annals of Pharmacotherapy, 43*, 700-706. doi: 10.1345/aph.1L522

Richter, S. (2018, April 13). *May 2018 Blackboard Learn upgrade – New features and cloud migration* [Blog post]. Retrieved from https://facdevblog.niu.edu/may-2018-blackboard-learn-upgrade

Roe, R. (2010). Design standards for safety warning messages in manuals: Increasing design saliency and adapting for a broader range of professional communication settings. *Graduate Theses and Dissertations,* 13342. Retrieved from https://lib.dr.iastate.edu/etd/11342

Rutherford, B. A. (2005). Genre analysis of corporate annual report narratives. *Journal of Business Communication, 42*(3), 349-378. doi: 10.1177/0021943605279244

Singh, M. K. S., Shamsudin, S., & Zaid, Y. H. (2012). Revisiting genre analysis: Applying Vijay Bhatia's approach. *Procedia - Social and Behavioral Sciences, 66*, 370-379. https://doi-org.ezproxy.lib.uwstout.edu/10.1016/j.sbspro.2012.11.280

Swales, J. M. (1990). *Genre analysis: English in academic and research settings.* Cambridge: Cambridge University Press

Tribble, C. (2001). Small corpora and teaching writing. In M. Ghadessy, A. Henry, & R. L. Roseberry (Eds.), *Small corpus studies and ELT: Theory and practice* (pp. 381-406). Amsterdam: John Benjamins.

TouchBistro. (n.d.). *What's new.* Retrieved from https://www.touchbistro.com/whats-new/

Yates, J., & Orilkowski, W. (2002). Genre systems: Structuring interaction through communicative norms. *The Journal of Business Communication, 39*(1), 13-35. https://doi.org/10.1177/002194360203900102

Zhou, S. (2012). 'Advertorials': A genre-based analysis of an emerging hybridized genre. *Discourse & Communication, 6*(3), 323-346. doi: 10.1177/1750481312446265

Appendix: Corpus Details

Release notes for the following companies and products were included in the corpus for this genre analysis.

Company and Product	**Industry**	**Year Company Was Founded**	**Approximate Number of Employees (Current)**	**Headquarters Location**
Accounting Seed Accounting Seed	Financial Management	2010	30	Columbia, MD
Blackboard Blackboard Learn	Education	1997	3,000	Washington D.C.
Cisco Meeting Server	Telecommunications	1984	74,200	San Jose, CA
Clock Clock POS	Hospitality	1994	30	London
Jamf Jamf Pro	Mobile Device Management	2002	700	Minneapolis, MN
TouchBistro TouchBistro	Restaurant	2010	225	Toronto