International Journal *of* Designs for Learning

2019 | Volume 10, Issue 1 | Pages 64-77

DESIGNING A REVISION TOOL TO DISTINGUISH SURFACE-LEVEL AND TEXT-BASED WRITING FEEDBACK

Kalianne L. Neumann[1] & Theodore J. Kopcha[2]

[1]*Oklahoma State University;* [2]*The University of Georgia*

This paper presents a design case that describes the design, development, and user experience testing of a Google Docs revision add-on. The add-on is an instructional, peer review tool intended to help students distinguish surface-level feedback from text-based feedback in order to develop their revision task schema. Eleven secondary teachers completed a survey about using the add-on for instructional purposes, and 56 secondary students completed a survey after using the tool to provide feedback to a peer's writing and make changes to their writing based on feedback provided to them through the tool. Thematic analyses revealed recommendations for modifications and additions to the tool. Next steps include researching the effects of the updated add-on on secondary students' revision task schema development.

Kalianne L. Neumann is an Assistant Professor of Educational Technology at Oklahoma State University. Her research focuses on the development of revision task schema as well as technology integration in K-12 settings.

Theodore J. Kopcha is an Associate Professor of Learning, Design, and Technology at The University of Georgia. He studies technology integration in K-12 and higher education, including technology-enhanced cognitive apprenticeships as well as problem- and project-based STEM learning.

https://doi.org/10.14434/ijdl.v10i1.24118

INTRODUCTION

Improving the writing skills of secondary students is a national concern. Despite the obvious need for writing skills in the attainment and completion of jobs in the workforce, many students do not exhibit proficient capabilities in writing. The Nation's Report Card for Writing in 2011 reported that 73% of twelfth graders exhibit basic or below basic ability in writing, which means they have partial mastery of knowledge and skills required for their grade level (National Center for Education Statistics, & Educational Testing, 2012). While new, Internet-based technologies afford instant and rapid publication of student writing, they also have encouraged a lack of revision in young writers (Verheijen, 2013). As it is, students tend to do little revision without the help of others (Scardamalia & Bereiter, 1986). This is problematic because research indicates that revision skills can impact student achievement starting in the upper middle grades (Limpo, Alves, & Fidalgo, 2014). It is important for students to understand what revision is and how to employ revision strategies in their own writing.

To better support students in understanding revision and adopting revision strategies in their writing, we (the designers/authors) developed a Google Docs add-on called *Revision Assistant*. The add-on was designed to support the formation of revision task schema among secondary students. According to Hayes (1996), a task schema is "a package of knowledge, acquired through practice, that is useful for performing the task and is retrieved as a unit when cues indicating the relevance of the schema are perceived" (p. 16). Hayes used this definition to identify three cognitive processes associated with revision task schema: text processing (i.e., critical reading), reflection (i.e., problem solving, decision making), and text production. Engaging in these cognitive processes is critical when revising because it not only improves writing quality but also supports the development of robust revision task schema (MacArthur, 2012). *Revision Assistant* was developed to encourage students to engage in these processes, both when making changes to their own writing and also when providing feedback to someone else's writing.

ORIGIN OF THE DESIGN

The design of *Revision Assistant* was inspired by prior research conducted with the first author's middle school Language Arts students in the Pacific Northwest (Neumann & Kopcha, in press). The first author's previous experiences as a high school and middle school English Language Arts teacher led her to observe that many students struggled with major aspects of formal writing: conventions of standard English (e.g., capitalization, punctuation) and clarity of ideas (e.g., content, organization). To test these observations and study the effects of peer and teacher feedback on the revision of argumentative letters in Google Docs, we conducted the previous study (Neumann & Kopcha, in press); both peers and the teacher left feedback in the form of Google Docs comments. The results indicated that peers focused on surface-level feedback. Surface-level feedback suggests changes to the conventions of standard English (e.g., pronoun/verb usage, capitalization, punctuation, spelling) that preserve the meaning of the writing, whereas text-based feedback suggests changes that alter the meaning of the writing by adding, omitting, and reorganizing text (Faigley & Witte, 1981). We hypothesized that the feedback focused on more complex text-based errors, such as organization and purpose, was lost in an overwhelming stream of comments about surface-level errors. This finding was not surprising of inexperienced writers; they tend to focus on providing surface-level feedback and making surface-level changes during revision because they already have a schema for these types of errors, and text-based errors are more challenging to identify and correct (McCutchen, 2006; MacArthur, 2013, 2016).

INITIAL DESIGN DECISIONS

Revision Assistant was developed from an initial set of design decisions. The first decision was practical in nature. Both designers/authors have over a decade of K-12 professional development experience and were former K-12 teachers—the first author in secondary English Language Arts, and the second author in secondary mathematics. As former teachers, we (the designers/authors) wanted to build a tool that would be readily and widely available to teachers. Rather than build a proprietary system, we decided that *Revision Assistant* should be built as an add-on for Google Docs.

This decision was based on the increasing popularity of current approaches to K-12 technology integration, such as Bring Your Own Technology (BYOT) and 1:1 Chromebook initiatives. Districts who adopt these approaches rely heavily on G Suite, a web-based suite of productivity tools offered for free by Google for Education (e.g., Google Docs, Google Slides).

Each of the remaining initial design decisions was not only based on our previous K-12 experience but also grounded strongly in the theory associated with the development of revision task schema. For example, the second decision was intended to make revision task schema more manageable for students. To do this, we sought to distinguish surface-level and text-based feedback. Our vision was to visually depict surface-level feedback, similar to proofreading marks, on the document so students would not overlook the text-based feedback provided through comments. The process of working with different types of feedback (i.e., surface-level vs. text-based) is directed by the revision task schema and involves critical reading of the feedback in the context of the writing. By instantly distinguishing the two types of feedback provided to them, users of *Revision Assistant* could critically read their writing and feedback to evaluate the suggestions from their peer and the effectiveness of those suggestions for their writing. This feature had the potential to improve student engagement in text processing because it helps activate editing schema (Hayes, 1996) and functions like a rubric or checklist (Ferretti & Lewis, 2013; MacArthur, 2013).

In addition to visually portraying surface-level feedback, we sought to engage users in the cognitive process of reflection, which is directed by the revision task schema and includes both problem solving and decision making. These two components are intricately linked; problem solving occurs when a writer detects a problem based on their goals but is unsure of how to correct it, and decision making occurs when a writer identifies solutions based on their evaluation and decides which solution to use to solve the problem (Beal, 1990; Hayes, 1996; MacArthur, 2016). To ensure that users engage in both the problem-solving and decision-making processes when using the tool, our third design decision was to exclude the 'accept' or 'accept all' feature that is commonly found in many word processing tools (e.g., Microsoft Word, Google Docs).

The decision to exclude an 'accept all' feature and require that users manually accept or reject was a result of our intention to help students internalize surface-level errors through repeated use. The previous research study supported the omission of the 'accept all' feature; students in that study tended to arbitrarily accept every suggestion provided to them without considering whether those suggestions were appropriate or made sense. If this were to happen with *Revision Assistant,* students would not learn from common errors made while writing because they would not have the opportunity to engage in the cognitive processes of problem solving and decision making. Thus, the omission of the accept all function forces users to make a decision about if and how to use the feedback provided to them, which could expand their revision task schema (Beal, 1990; Hayes, 1996; MacArthur, 2016). Additionally, since surface-level errors are often the focus of a beginning writer's revision task schema, this feature supported the production of surface-level changes (Hayes, 1996; McCutchen, 2006; MacArthur, 2016).

Below we describe each design decision in the context of the four phases of development of *Revision Assistant*. Within each phase there were emergent design decisions, which are explained in each of the phases.

Phase 1: Initial Design

Phase 1 involved prototyping user behaviors using mock-ups of the tool that were hand-drawn in a notebook (see Figure 1). The design depicts a toolbar at the top of a Google Doc; the toolbar includes images of handwritten proofreading marks, which were to appear on the page when pushed by the user. After developing the initial design on paper, the look and functionality of the tool were mocked up using images created using Paintbrush for Mac. As shown in Figure 2, a user would use the add-on by manually inserting feedback. This action would be completed by highlighting the location for the feedback and then pushing the necessary button. The images were then integrated into a brief video clip (~5 min) that demonstrates the intended functionality of the tool (see video, Figure 3).

One design decision that emerged during this phase was the addition of an export comments feature. The export comments feature was inspired by the academic revision process and the need to explain to reviewers how a comment was or was not addressed in the revision. The intended functionality of the feature was to export the comments made throughout a given document in table form and provide a space for users to explain their changes. This would allow a teacher or student to analyze those comments collectively and see what types

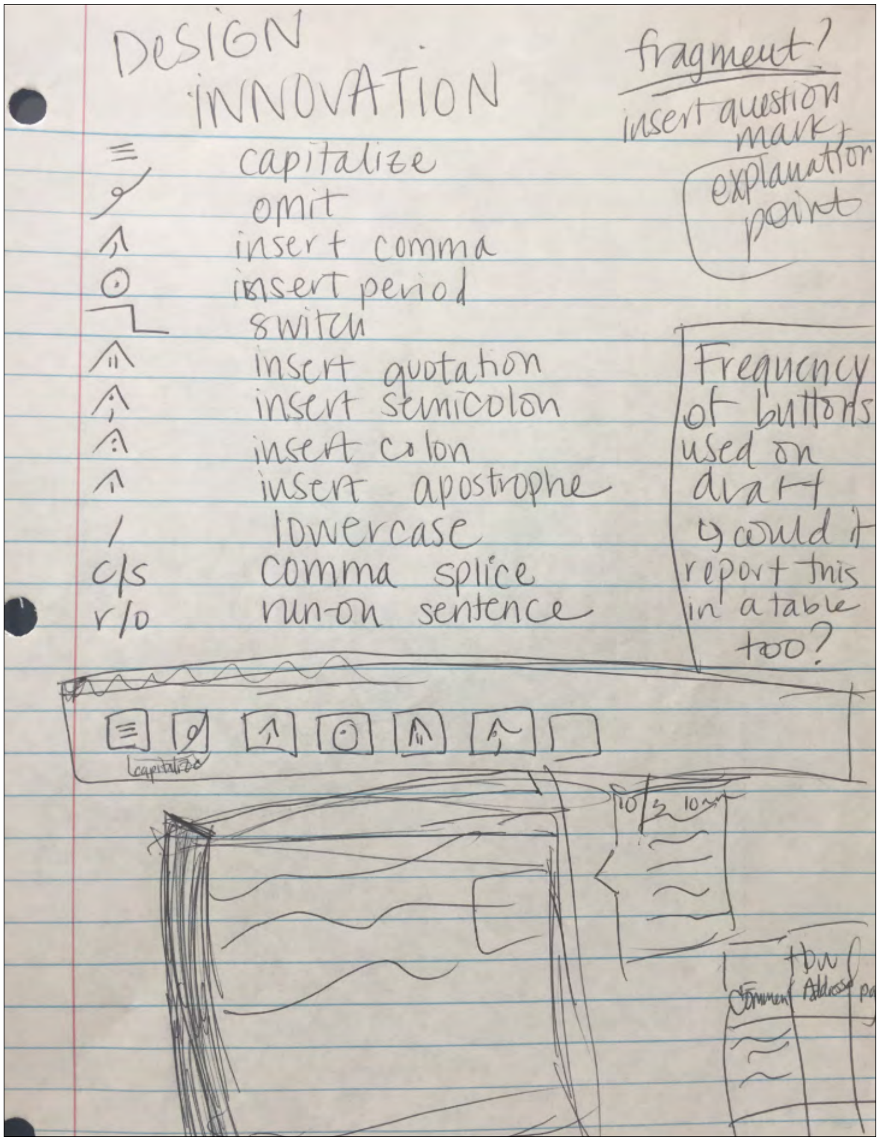
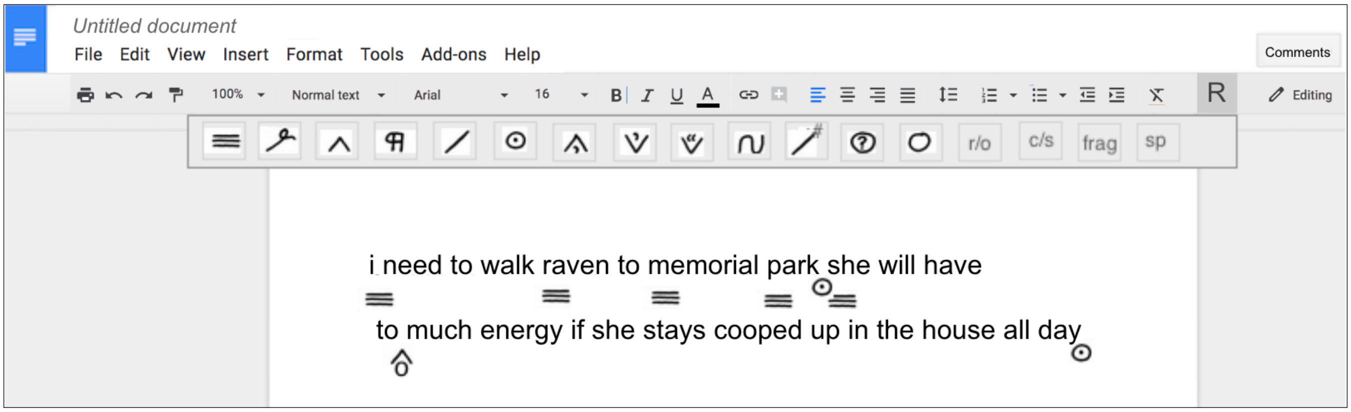
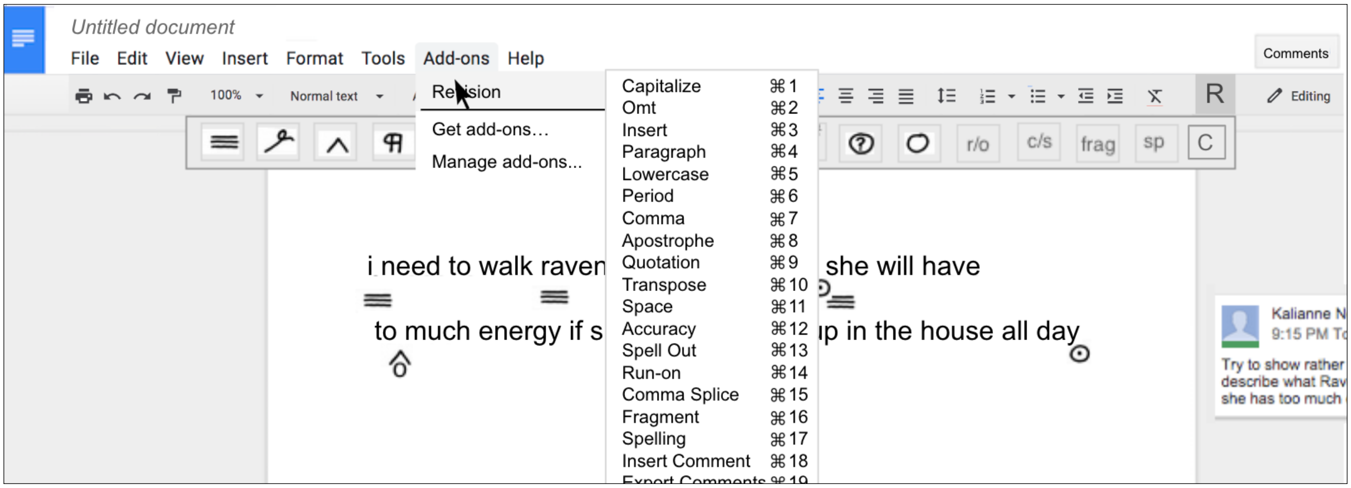


FIGURE 1. Initial hand drawn design for *Revision Assistant*, September 2014.



i need to walk raven to memorial park she will have to much energy if she stays cooped up in the house all day

FIGURE 2. Graphical image of the initial design, October 2014.



i need to walk raven … she will have

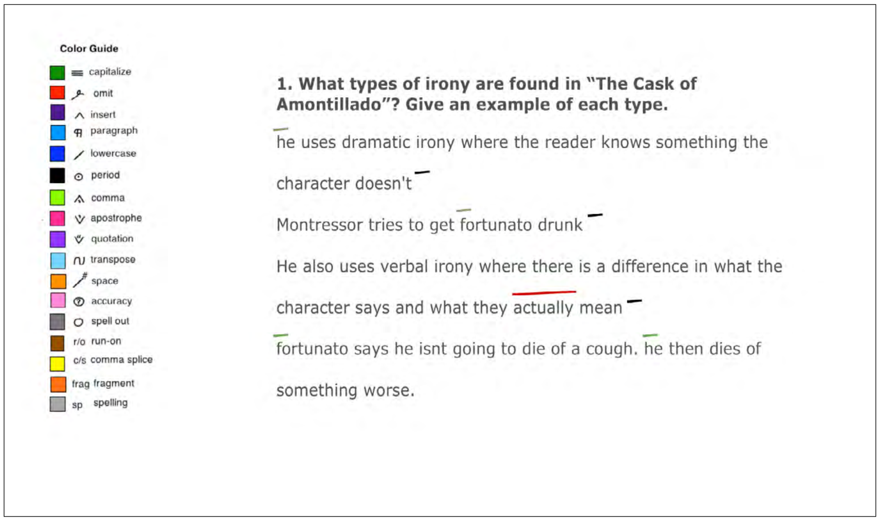
to much energy if s … p in the house all day

Kalianne N
9:15 PM To

Try to show rather
describe what Rav
she has too much

FIGURE 3. Frame taken from a video clip depicting the intended functionality of the initial design, November 2014. The video clip is archived and accessible at: https://purl.dlib.indiana.edu/iudl/media/821g550g62

of content-based errors were made most often. We anticipated that this design decision would better support all three of the cognitive processes (e.g., text processing, reflection, text production) because students could see a comprehensive list of the text-based errors made (De la Paz, Swanson, & Graham, 1998), set and evaluate goals for their revisions (Rijlaarsdam, Couzijin, & van den Bergh, 2004), and justify their solutions to those errors (Berzsenyi, 2001).



1. What types of irony are found in "The Cask of Amontillado"? Give an example of each type.

he uses dramatic irony where the reader knows something the character doesn't

Montressor tries to get fortunato drunk

He also uses verbal irony where there is a difference in what the character says and what they actually mean

fortunato says he isnt going to die of a cough. he then dies of something worse.

FIGURE 4. Color guide sidebar and example of text with overline suggestions, November 2014.

Phase 2: Appearance Compromises

In Phase 2, we began working with a university-based computer programmer to develop a working prototype of the design. The meetings we had with the developer revealed a major logistical flaw in the initial design: the proofreader's marks could not be inserted on the page due to issues related to spacing and formatting. Using images to depict each error symbol required the use of a double-spaced document; those images would not appear correctly on a single-spaced document. However, the developer noted that there was no way to program a Google add-on to control the line spacing within a given user's document. We therefore modified the design. Instead of the symbol actually appearing on the page, the new design associated a color with each of the errors/buttons, and the color would place a line over the necessary letters or words to indicate the location of feedback. Additionally, we designed a sidebar that would function as a key for the colors associated with each proofreading mark (see Figure 4). We believed that this modification to the design would maintain our goal of visually distinguishing surface-level and text-based errors.

The revised design (i.e., using colors instead of proofreading mark images) was then mocked up and tested with a group of six secondary English teachers in Georgia. The mockup consisted of a series of still images developed in Paintbrush. We printed out an image of each of the major design features (e.g., toolbar, color key, sample text) and arranged the images on a table as if they were a functioning piece of technology. The teachers physically touched the images with their finger to model their user behaviors and used a marker to draw a line over the current text at the location of the identified error. For example, they would first touch the symbol of the feature they wanted to use, then locate a pen or marker that represented the color associated with the button, and finally draw a line over the appropriate letters or word before setting down the pen or marker and moving on to their next suggestion. After providing all of their surface-level feedback, the teachers wrote comments using sticky notes and placed them on the table near the location of the comment. When all comments were written, the teachers pushed the export comments button with their finger and relocated the sticky notes to a printed copy of the export comments table.

After reflecting on their process and considering how it would function in their classrooms, the teachers indicated that they liked how receiving feedback would require a student to make a decision about that feedback. However, they indicated a strong preference for displaying the proofreading marks as symbols on the page. Although they understood the programming constraints, their concern was over the use of colored lines to indicate student errors. The teachers suggested that if the images could not appear on the page, the colors needed to highlight the letters or word instead of overlining them; they believed that students would ignore the colored overline just as much as they ignore the red spelling underline. After seeing what the overline looked like on paper, we agreed with the teachers. The last thing we wanted was another function that students ignore.

While we also still preferred to have the proofreading marks on the page, the logistical constraints led us to update the design so that the buttons highlighted the letters and words, which we believed would bring more attention to a suggested change and improve the tool's ability to visually distinguish surface-level errors (see Figure 5).

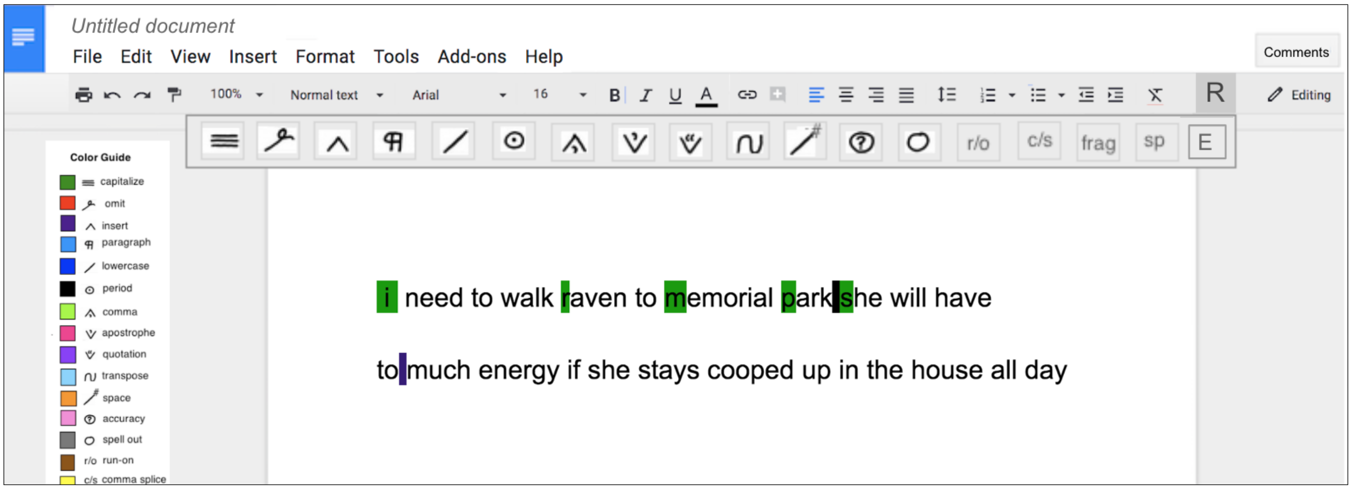
Throughout Spring 2015, we continued to work with the university-based developer and decided to add a feature that would track the frequency of proofreading buttons used on a document. This decision supported one of our initial design decisions to support problem-solving and decision-making processes: intentionally omitting an 'accept all' feature. We believed that tracking the frequency of students' use of proofreading buttons would activate their editing schema (Hayes, 1996), would function like a checklist for processing surface-level errors (Ferretti & Lewis, 2013), and, over time, could help a student recognize common errors made in their writing and expand their revision task schema.

Phase 3: A Functional Add-On

Our work with the developer stalled as Summer 2015 approached, and we were left looking for a new programmer to help us transform our paper prototype into a working prototype. An instructional technologist from the local school district put us touch with the district's Google for Education contact, who connected us with Promevo. Promevo is a Google Premier Partner that specializes in Google solutions for business and education institutions. They offer Chrome hardware, G Suite licenses, Chrome management tools, and custom app creation. Promevo agreed to develop *Revision Assistant* on a pro bono basis to further explore app design within the Google for Education environment.

We verbally pitched the idea to Promevo on July 21, 2015, and the following day, we sent the company a newly updated version of the video depicting the design and functionality of the tool. The updated video combined features from Phase 1 and Phase 2; this emphasized the following features: (1) proofreading symbols as images rather than color coding, (2) an export comments table, and (3) a frequency tracker of toolbar buttons (see Figure 6). Based on teacher feedback and a review of the literature, we believed that these features were the most important to include in the design because of their potential to support the expansion of a revision task schema.

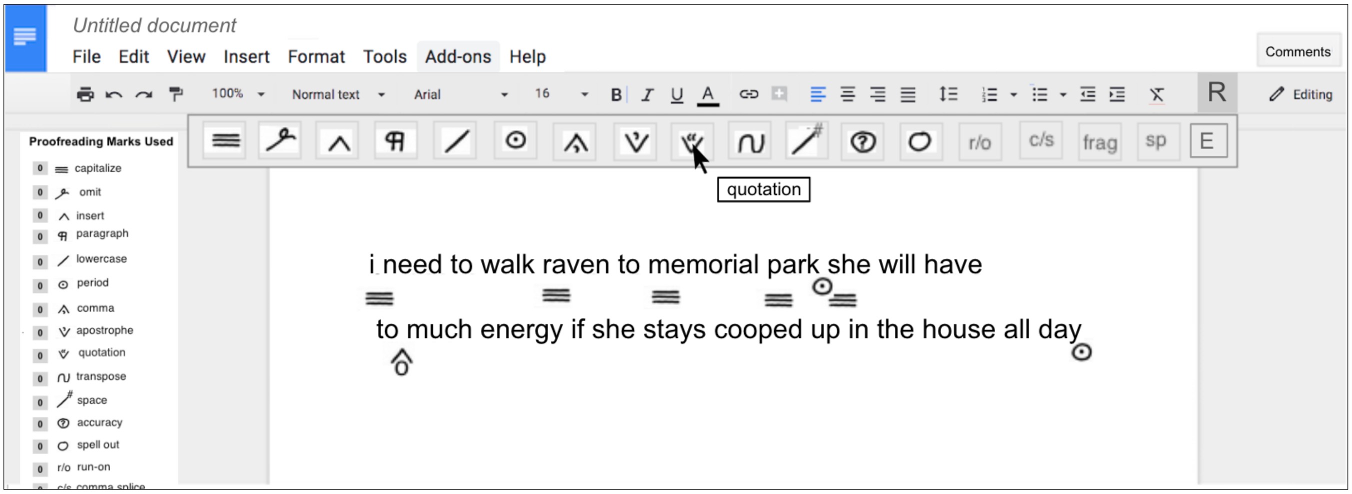
By August 2015, Promevo created a working prototype of *Revision Assistant*. Several features did not appear in the add-on as we had hoped, so we met again with the developers to discuss the working prototype. One issue was that the proofreading marks did not appear as images in the document. The Promevo programmers confirmed that it was not feasible to integrate the proofreading mark images on the



i need to walk raven to memorial park she will have

to much energy if she stays cooped up in the house all day

FIGURE 5. Image of the design decision to include colors that highlight occurrences and a color key instead of proofreading mark images, December 2014



i need to walk raven to memorial park she will have to much energy if she stays cooped up in the house all day

FIGURE 6. Frame taken from a video clip depicting the updated design that was sent to Promevo, July 2015. The video clip is archived and accessible at: https://purl.dlib.indiana.edu/iudl/media/d375840r1m

page due to the limitations of Google's API. However, they designed a solution to this issue that we had not previously conceptualized. After opening the add-on, a user would see a new toolbar along the right side of the Google Doc that contained a total of 23 buttons. The majority of buttons dealt with issues related to writing conventions, such as correcting case format (e.g., capitalization, lower case) or inserting punctuation marks (e.g., comma, colon, period). When used, these proofreading marks were set apart from normal text in two visually distinct ways. First, they were surrounded by parentheses at the location of the user-identified error. Second, they were set to a red text color. For example, an inserted comma appeared in red font as this: (,). While we (the designers/authors) and teachers who tested the paper prototype preferred the idea of using proofreading marks to indicate an error, our collective emphasis was that the tool visually distinguished surface-level and text-based errors to improve text processing of these errors. The solution created by the Promevo programmers was not our ideal solution (i.e., proofreader's marks); however, it was a more powerful solution than our previous color-coding and color code key solution. Promevo's solution not only did a better job of distinguishing error types but also of using the mark outputted on the page to indicate the actual suggested change rather than a color linked to a key. In a sense, the solution ended up creating a new version of the handwritten proofreader's marks.

After the meeting, we spent some time exploring the features of the add-on and came up with four main design improvement suggestions for the programmers. The first was that the toolbar was cumbersome; having 23 different buttons made it difficult to easily find and make use of the various proofreading marks that were available. To address this, we suggested organizing the buttons into two categories to make scrolling through them more manageable and appealing for secondary students. We suggested the categories of Conventions (15 buttons) and Structure (7 buttons) and that they function as expandable dropdowns with all buttons in each category appearing in alphabetical order.

The second thing we noticed was the abrasive nature of the color red used to indicate all of the surface-level suggestions. To remedy this, we suggested that buttons included in the Conventions category be colored blue, with the exception of the Omit button, and Structure buttons be colored green. We believed that creating two categories (i.e., Conventions and Structure) and using a wider variety of colors to indicate error types would encourage students to engage in text processing by activating editing schema (Hayes, 1996) and functioning like a checklist for different types of errors (Ferretti & Lewis, 2013; MacArthur, 2013).

Our third suggestion was to add an occurrence tracker on the toolbar to calculate the number of times each button was used on the document. This had the purpose of not only promoting text processing of surface-level errors (Ferretti & Lewis, 2013) and informing students about common errors they make while writing but also informing teachers of common errors students make while writing (Hayes, 1996).

Finally, our fourth suggestion was to add an export comments button that exported all comments on a document into another Google Doc or Google Sheet. The goal of this feature was to allow both students and teachers to examine all documents on a page in an organized table. Providing this type of organized list of errors had the potential to engage students in text processing of text-based errors (De la Paz et al., 1998), setting and evaluating goals for their revisions (Rijlaarsdam et al., 2004), and justifying their solutions to those errors (Berzsenyi, 2001).

Our suggested changes were quickly conceptualized. The Promevo programmers were able to organize the toolbar into two categories: Conventions and Structure, and all buttons within each category were alphabetized (see Figure 7). The programmers also made it so Conventions buttons output blue color, with the exception of red output for the Omit button, and Structure buttons output green color. All buttons were tracked on the toolbar by the number of times they occurred on the page (see Figure 7). Finally, the Promevo programmers added an Export Comments button that exported all comments on a document into a table at the end of the document. Comments were exported by timestamp and included the name of the person making the comment in the first column and the text of the comment in the second column. While we knew that we would eventually want to make additional updates, we submitted *Revision Assistant* to Google for final approval. On September 30, 2015, *Revision Assistant* was released as a free Google Doc add-on (see Figure 8), and it has over 39,000 users in less than three years.

Phase 4: Users' Experience of the Design

Once a working version was available for use, we conducted design research to test the functionality and usability of *Revision Assistant* with its intended audience: secondary teachers and students. We surveyed 11 secondary teachers who used the add-on and integrated it into their classrooms, and 56 secondary students (grades 6-9) who used the add-on to conduct peer review and revise their writing with feedback provided to them through *Revision Assistant*.

Secondary Teachers

All of the secondary teachers who completed the survey indicated that *Revision Assistant* was easy to use, it supported the revision process, and they would recommend it to other teachers; however, they also revealed that they did not always know where to find buttons, the colors did not always distinguish where changes needed to be made, and that making changes was sometimes difficult. After integrating the tool into their instruction for peer review, all of these teachers believed that their students understood what to do with the tool, used the tool effectively, provided useful feedback, and used the feedback they received to revise their own writing. Finally, the majority of teachers indicated that they used both the exported comments and occurrence tracker to inform their instruction.

Secondary Students

Secondary students engaged in two primary activities when using *Revision Assistant*. The first was providing feedback to another student. Students were generally positive about providing feedback to another student because it was easy and saved time that would otherwise be lost writing comments. While students found this helpful for providing suggestions of errors to look for, they at the same time noted how they were sometimes confused about how the toolbar worked.

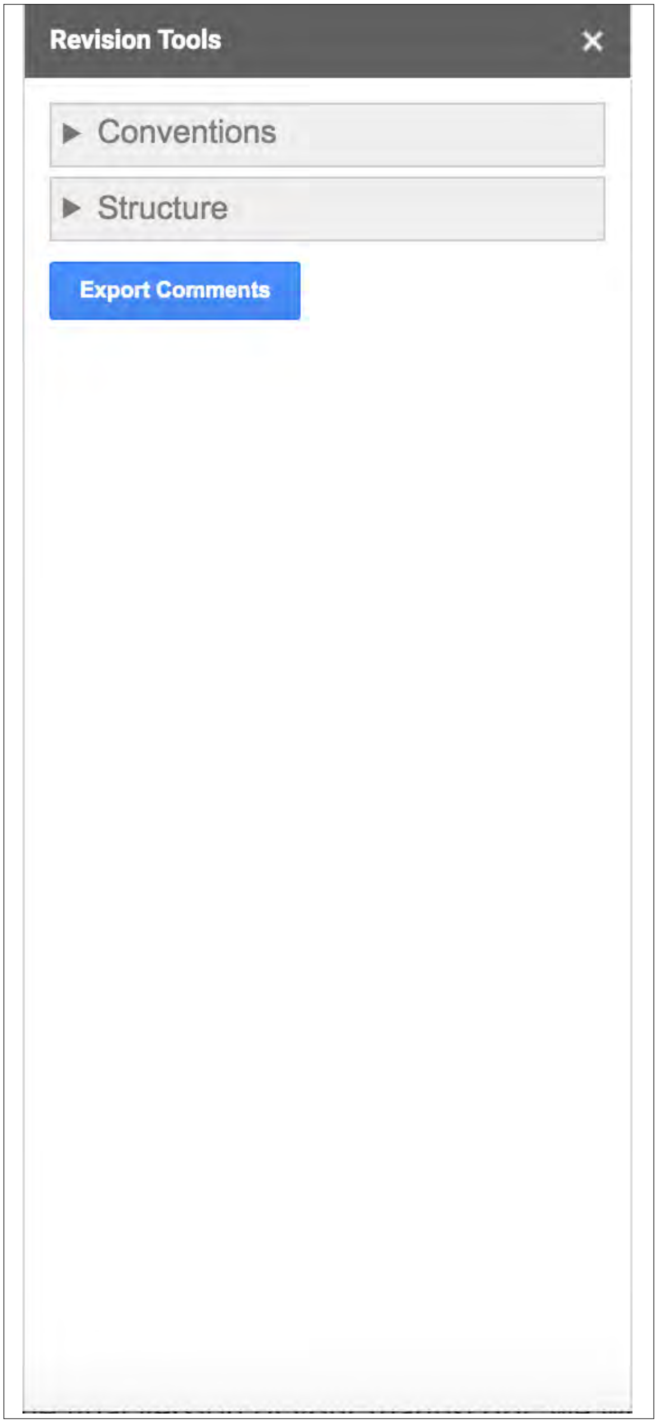
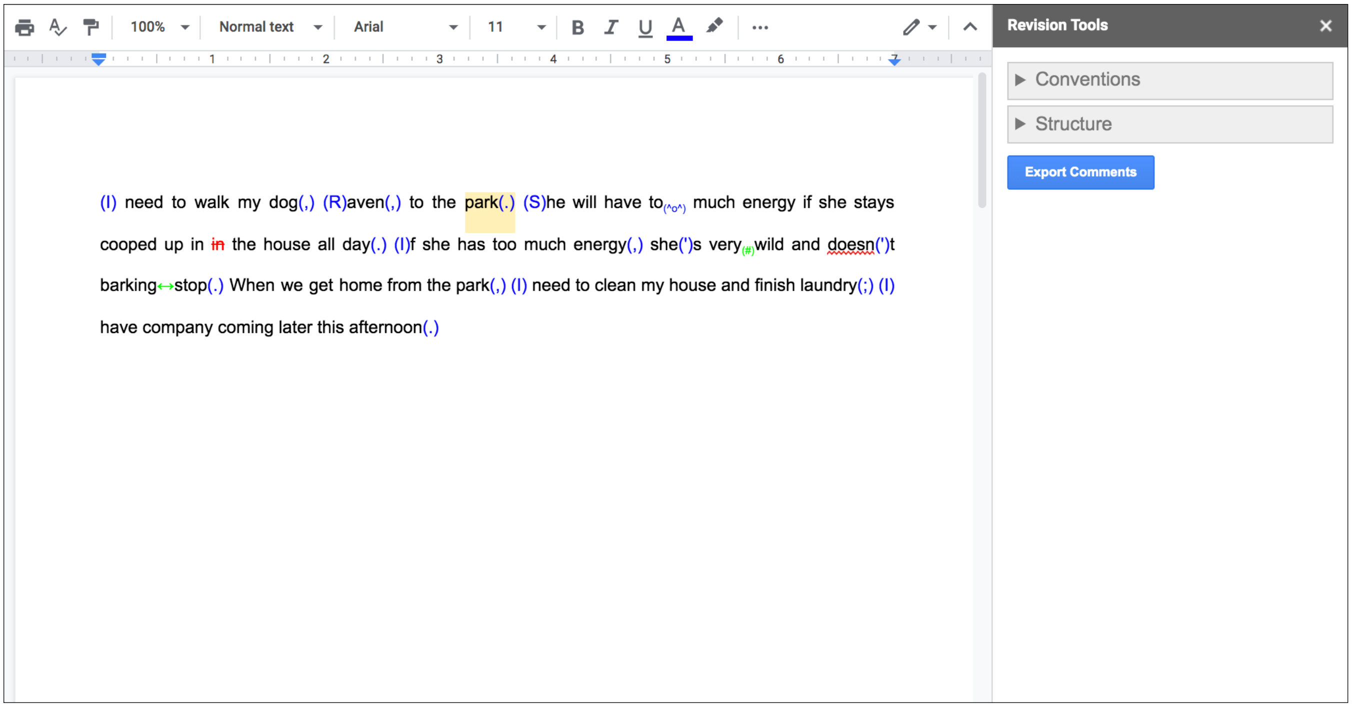


FIGURE 7. Frame taken from a video clip depicting the toolbar categories and expanded categories in the published version of *Revision Assistant* (.gif format). Created from "Revision Assistant" by Promevo, 2015 (https://chrome.google.com/webstore/detail/revision-assistant/aalcpknmagohgaakdjhapkggbcaidepa?hl=en-US). Adapted to .gif format with permission. The video clip is archived and accessible at: https://purl.dlib.indiana.edu/iudl/media/g74q57qm0m



(I) need to walk my dog(,) (R)aven(,) to the park(.) (S)he will have to$_{\text{(^o^)}}$ much energy if she stays cooped up in ~~in~~ the house all day(.) (I)f she has too much energy(,) she(')s very$_{(\#)}$wild and doesn(')t barking↔stop(.) When we get home from the park(,) (I) need to clean my house and finish laundry(;) (I) have company coming later this afternoon(.)

FIGURE 8. Frame taken from a video clip demonstrating how a user would give feedback to a paragraph with *Revision Assistant* and export comments into a table at the end of the document (.gif format). Created from "Revision Assistant" by Promevo, 2015 (https://chrome.google.com/webstore/detail/revision-assistant/aalcpknmagohgaakdjhapkggbcaidepa?hl=en-US). Adapted to .gif format with permission. The video clip is archived and accessible at: https://purl.dlib.indiana.edu/iudl/media/019s066329

Overall, when students were giving feedback to a peer, they found that the color distinctions between types of feedback were not helpful, were frustrated that the occurrence tracker only added occurrences, and lacked understanding of the meaning of many of the toolbar buttons.

The second action that students used *Revision Assistant* for was to receive feedback from a peer. Students were generally positive about receiving feedback because it was easy to make changes, feedback was visual in nature, and feedback—both surface-level and text-based—was organized. However, students also found some aspects of receiving feedback problematic. Several of these were a result of mechanical issues with the tool. To begin, the colored font remained when making changes, forcing students to change it back after editing. Students also found it difficult to find suggested feedback on a document, and the occurrence tracker was not sophisticated enough to reduce an occurrence whenever a student made appropriate changes. Finally, students noted that there was not enough information in the comment table; they expected to see information about the context of the comment, such as the text associated with the comment or the page number. While their experiences using *Revision Assistant* varied, the majority of students indicated that they would use the tool again in the future.

DESIGN FAILURES AND REVISIONS

Despite the generally positive feedback we received from secondary teachers and students, both groups highlighted design failures and suggestions for improvement in the next version of *Revision Assistant*. Table 1 depicts the design decisions up to Phase 4, failures of those design decisions identified by users, and updated design decisions based on those failures. The only design decision that is not included in the table is the first one (building a tool that would be readily and widely available to teachers) because neither teachers nor students experienced failures with this aspect of the design. To begin, secondary teachers requested a variety of additional buttons. The requested buttons included: word choice, verb tense, subject-verb agreement, noun-verb agreement, pronoun agreement, vary sentences, simple sentence, compound sentence, complex sentence, and compound-complex sentence. While these buttons would enhance peer review activities, teachers saw that they also supported another use for *Revision Assistant* that was not part of the original design. Because *Revision Assistant* was a Google add-on, teachers could easily check student understanding of grammar by distributing a model text using Google Classroom and having students label concepts of conventions in those texts. Secondary teachers also suggested the toolbar be reorganized into three main categories: Mechanics, Usage, and Sentence Formation. They felt these three categories would better align with how they discuss surface errors with students.

Both the teachers and students identified functional issues with the exported comments table and occurrence tracker that made them difficult to use effectively. The teachers noticed that comments were exported by timestamp, which became problematic when a student provided comments in a nonlinear fashion; because of this, they requested that comments be exported in the order in which they appear on the page. Teachers also requested the addition of a blank column that would allow them to request student explanations of the changes made based on the comment provided to them. Both secondary teachers and students explained that the comment table forced them to go back and forth between the text and comment because they did not know what part of the text a comment was associated with. As a result, we realized the necessity of additionally exporting the text associated with the comment into the table. The occurrence tracker frustrated secondary students because they were unable to take away an occurrence if they accidentally pushed a button, and while this did not bother teachers, the teachers wondered if the occurrences could also be exported into a table at the end of the document.

While the secondary teachers provided suggestions for making *Revision Assistant* a more effective instructional tool, secondary students highlighted features and provided suggestions for making *Revision Assistant* more useful for them. Several students noted that the green color output by the Structure buttons was too bright and difficult to see. Many of the younger students indicated that they were not fluent in the names of conventions of standard English or how to use them in writing; for them, accepting or rejecting *Revision Assistant* feedback was difficult because they were not confident in the appropriate ways of correcting errors in conventions. These students requested additional information about the buttons, such as a picture of the mark output by the button, definitions of each mark, and examples of how to use the mark correctly in writing, so they could make more informed decisions when accepting or rejecting feedback.

UPDATING THE DESIGN

After identifying the design failures and potential solutions to those failures, we attempted to address the solutions through trial and error with a copy of the Google Script. We were successful in updating the script to reflect the majority of suggestions from secondary teachers and students (see Table 1; Figure 9). Specifically, we added teacher-requested buttons and reorganized the toolbar into three categories (Mechanics, Usage, Sentence Formation) to improve engagement in text processing of surface-level errors. We created hover-over definitions and examples of each button and inserted an image of the mark output on each button to improve engagement in the reflective processes of problem solving and decision making when addressing surface-level errors. To improve text processing of and schema activation for Sentence Formation errors, we updated the green color output, so it was easier to visually identify. In the export comments table, we created a third column (Addressed? How?) to improve engagement in text production by providing a designated space to explain decisions and changes made during revision. After completing these updates, we sent the updated script and remainder of potential solutions to the identified design failures to Promevo. While they were not able to change the fact that comments appeared in timestamp order, the information included in the export comments table, or the ability to take away an occurrence, the Promevo programmers were able to create an export occurrences button and table that exports all occurrences of buttons used on a document into an organized table at the end of the document. The addition of this feature promotes engagement in text processing of common surface-level errors made by visually appearing like a checklist that summarizes all surface-level errors identified within a document.

Bump in the Road

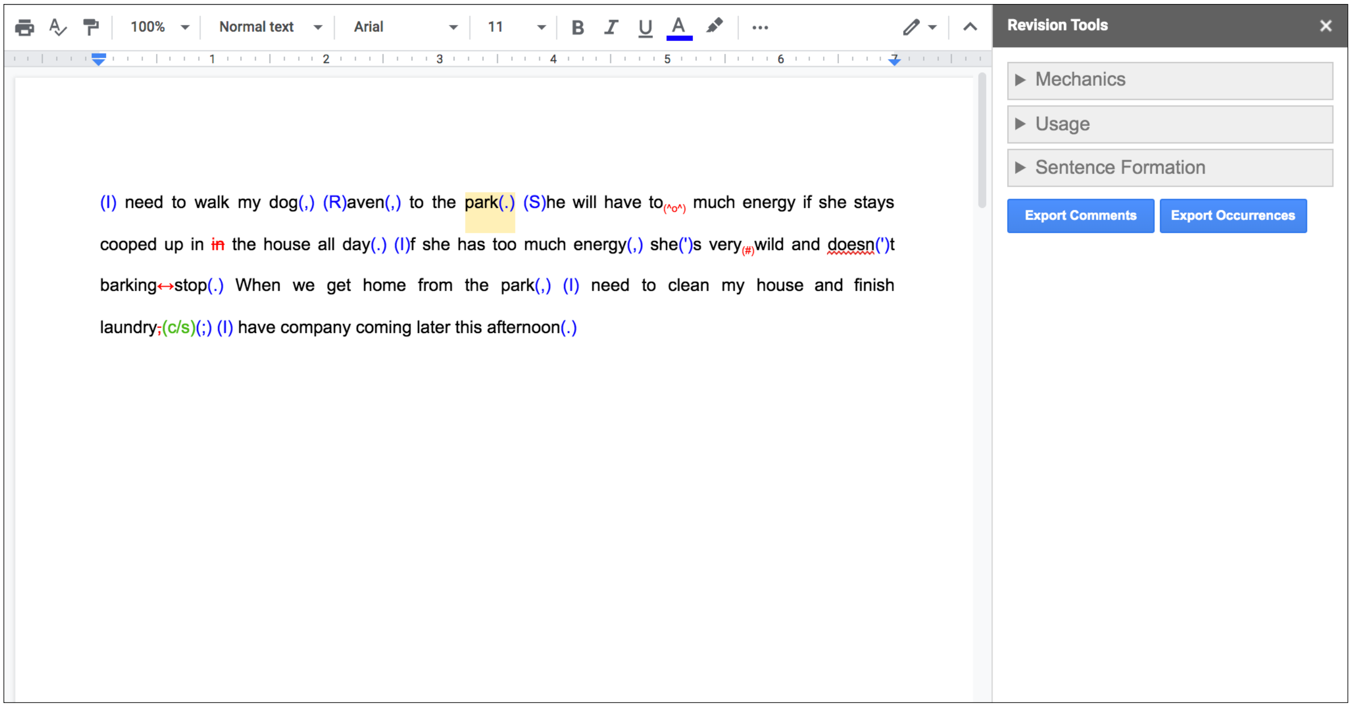
Once the script was ready to release as an update to the over 35,000 users we had at the time, we hit a major bump in the road that stalled the release of the updated version of the tool. An error message occurred when we attempted to deploy the add-on as a web add-on to the Google Chrome Web Store. Despite our attempts to contact Google and post on developer forums, we were unable to find a solution that would release the update to our current users. This major bump resulted in the publication of an entirely new add-on: *Revision Assistant, too*. The new add-on is searchable through the Google Chrome Web Store add-ons section, includes all of the previously mentioned updates, and updates the previous logo and promotional images with the squared symbol in the corner to distinguish the two distinct versions of the tool. The new add-on was a less-than-ideal scenario from a user perspective, as it forces users to install a new add-on rather than updating the existing add-on automatically. However, having two versions of the add-on is beneficial in some ways - creating two distinctly separate versions of the tool may be helpful for future research about functions afforded by the tool and their effects on student performance. Variations in toolbar design (i.e., two vs. three categories) and exported comment functionality can be compared to see which version better supports the formation of revision task schema.

DESIGN DECISIONS UP TO PHASE 4	INITIAL RATIONALE FOR LEARNING EFFECTS OF DESIGN DECISION	FAILURE BASED ON USER EXPERIENCE DURING PHASE 4	UPDATED DECISION / REVISION	ANTICIPATED LEARNING EFFECTS OF UPDATED DECISION / REVISION
Visually separate error types (i.e., surface-level / text-based)	Encourages students to engage in text processing of their writing and the feedback provided to them by activating editing schema (Hayes, 1996) and functioning like a rubric or checklist (Ferretti & Lewis, 2013; MacArthur, 2013).	Two types of errors were not enough; teachers wanted more.	Group errors by common type that aligns with language used by teachers: Mechanics, Usage, and Sentence Formation (see Phase 3).	Makes additional improvements to identifying and evaluating (i.e., text processing) surface-level errors by reorganizing the toolbar around the types of errors most commonly seen in the classroom.
			Color code each error type: Mechanics-blue, Usage-red, and Sentence Formation-green.	
		Teachers sought specific button types that were not initially included.	Addition of 10 new buttons: word choice, verb tense, subject-verb agreement, noun-verb agreement, pronoun agreement, vary sentences, simple sentence, compound sentence, complex sentence, and compound-complex sentence.	
Intentional omission of an 'accept all' feature	Engages students more deeply in problem solving and decision making when revising by forcing them to engage with feedback (Beal, 1990; Hayes, 1996; MacArthur, 2016). Also engages students in text production of surface-level changes (Hayes, 1996; McCutchen, 2006; MacArthur, 2016).	Students struggled to make effective decisions and revisions because they were not fluent with the names of conventions of standard English or how to use them in writing.	Button includes an image of the mark output when a user pushes the button (e.g., Comma (,)).	Improves engagement in the reflective processes of problem solving and decision making when addressing surface-level errors by providing examples and definitions to support both problem solving and decision making.
			Button includes a hover-over definition and examples of how the mark is used.	
Use colors to distinguish error types	Encourages students to engage in text processing of their writing and the feedback provided to them by activating editing schema (Hayes, 1996) and functioning like a rubric or checklist for different error types (Ferretti & Lewis, 2013; MacArthur, 2013).	Students indicated the green color was too bright.	Change the green color to darker shade - from #00ff00 to #4cbb17.	Improves text processing of identified Sentence Formation errors by making them easier to identify, which will activate editing schema for Sentence Formation errors.

TABLE 1. Failures of design decisions indicated through user experience, updated design decision, and learning effects.

DESIGN DECISIONS UP TO PHASE 4	INITIAL RATIONALE FOR LEARNING EFFECTS OF DESIGN DECISION	FAILURE BASED ON USER EXPERIENCE DURING PHASE 4	UPDATED DECISION / REVISION	ANTICIPATED LEARNING EFFECTS OF UPDATED DECISION / REVISION
Exported table of comments	Engages students in text processing as they view a comprehensive list of text-based errors (De la Paz et al., 1998), set and evaluate goals for their revisions (Rijlaarsdam et al., 2004), and justify their solutions to those errors (Berzsenyi, 2001).	Teachers requested an additional column for the comments table.	Create an additional column in the table labeled "Addressed? How?"	Improves engagement in text production by providing a designated space for students to explain decisions and changes made during revision.
		Teachers and students indicated that the comments table lacked critical information about the context of the comment.	Unable to resolve due to limitations of the programming language.	
		Teachers requested comments be exported in the order they appeared on the page instead of timestamp order.		
Occurrence tracker on toolbar	Engages students in text processing of surface-level errors made and provides a checklist for those errors (Ferretti & Lewis, 2013), which could help them recognize common errors made over time, while also activating editing schema (Hayes, 1996).	Teachers sought to create an exported occurrences table.	Create an export occurrences button that exports occurrences into a table at the end of the document.	Improves engagement in text processing of common surface-level errors made by visually appearing as more of a checklist that summarizes all surface-level errors identified.
		Students were frustrated that the occurrence tracker did not remove an item when a button was mistakenly pushed and removed.	Unable to resolve due to limitations of the programming language.	
		Teachers requested comments be exported in the order they appeared on the page instead of timestamp order.		

TABLE 1 (CONT.). Failures of design decisions indicated through user experience, updated design decision, and learning effects.



(I) need to walk my dog(,) (R)aven(,) to the park(.) (S)he will have to(^o^) much energy if she stays cooped up in ~~in~~ the house all day(.) (I)f she has too much energy(,) she(')s very(#)wild and doesn(')t barking↔stop(.) When we get home from the park(,) (I) need to clean my house and finish laundry~~;~~(c/s)(;) (I) have company coming later this afternoon(.)

FIGURE 9. Frame taken from a video clip depicting the reorganized toolbar, hover-over definitions and examples of each button, images of the mark output by each button, additional column in the exported comments table at the end of the document, and exported occurrences table at the end of the document. The video clip is archived and accessible at: https://purl.dlib.indiana.edu/iudl/media/b29b19gw1p

REFLECTING ON FAILURE

Our initial decision to create a revision tool as an add-on for Google Docs led to a number of unanticipated constraints due to the limitations of Google's programming interface. We ultimately had to move away from our original vision of using images to represent proofreading marks and instead used color-coding and parentheses to display those marks visually. This led to several compromises during the design and development of *Revision Assistant* and, at times, failures with our design evidenced during our user experience. This begs the question - if we had it to do over again, would we make the same choice? That is, would we have chosen to create our tool using Google's programming language if we knew the issues it would create?

After having some time to reflect on the user experience, we feel that the compromises made due to our choice of platform were not the primary cause of our major design failures. For example, teachers wanted more buttons and wanted them organized differently than we originally anticipated. This issue is more a result of the ways theory and practice connect than the technological platform. The literature on revision discusses two types of errors—those that are surface-level and those that are text-based. We based our initial design on this dichotomy. Yet teachers have a more nuanced perspective of revision, further separating those errors into three categories so that students can more easily differentiate among them and master them over time. Similarly, students could not effectively make use of many of the buttons because they were not fluent with all the error types in the tool. While unanticipated, both issues were not a shortcoming of the platform but rather of our understanding of the theory and practice around revision.

In the end, only two issues remained unresolved due to the limitations of Google's programming interface - these deal with the occurrence tracker and the exported comments. At this time, the occurrence tracker does not remove an occurrence of an error when that error is resolved. Likewise, the exported comments appear without reference to the actual text, making it hard at times to understand the nature of some comments. Both were important analysis features of our initial design in that they provide critical information about the number and types of errors made and resolved during peer review. This was a disappointing trade-off for us as designers - in order to offer our tool on a popular, widely-used platform (i.e., Google) we had to settle for creating analysis features that were less-than-optimal.

At the same time, we are pleased that the analysis features still provide information that helps analyze student errors in a general sense. While they lack the specificity needed to engage in deeper analysis, students and teachers can use them to see patterns of error-making and address them accordingly. This outcome remains strongly aligned with the spirit of our initial design in that we sought to provide teachers and students with a tool that helped support the cognitive process of reflection on writing. With this in mind, we feel that the benefits of working within the Google platform continue to outweigh the drawbacks created with our initial vision for our analysis features.

IMPLICATIONS FOR DESIGNERS

For designers looking to work with others to develop educational technologies, our experiences designing and developing *Revision Assistant* and *Revision Assistant, too* have several implications. First, it is important to be able to articulate the ideal conception—the one they would make in a world without constraints—of the educational technology they are designing. Intentionally ignoring perceived or known constraints could help designers and others assisting with development make the original vision come to fruition or create an alternate solution that was not previously identified. We experienced this when we transitioned from working with a university-based programmer to working with Promevo. Despite knowing that inserting the proofreader's marks on the page was not likely a realistic possibility, we sent Promevo a video of our ideal vision of the tool that used the proofreader's marks, and they created an alternative solution that not only maintained a core principle of our design but also provided a better solution than our previous alternative solutions.

A second implication is for designers to identify the core principles that are at the heart of their design; these principles should be the ones that they value the most—the ones they would not be willing to make compromises about. For us, the core principles were (1) to visually distinguish surface-level and text-based errors and (2) to be created for the Google platform. While it may have visually appeared different in each phase, both of these principles were present throughout each iteration of our design and development.

Finally, designers should understand that compromising aspects of the ideal version of their educational technology is likely a reality; this will help designers devise strategies to persevere through the challenges, delays, and different iterations of the tool. Early on in our design and development process, we were faced with the compromise of using colored lines over identified errors instead of the proofreading marks, which was not ideal. Testing a paper prototype of this solution with teachers confirmed that this was a weak feature of the tool's design. Our testing with teachers provided a suggested solution for the design and allowed us to collect valuable data on our paper prototype.

CONCLUSION

Now that the tool is updated to include all current, feasible solutions, we are shifting our focus from design research that examines whether the tool works as intended to research that examines the tool's impact on student performance. Specifically, we are studying secondary students' repeated use of *Revision Assistant, too* to investigate whether repeated use of it results in the intended effects. We imagine that after repeated use of the tool, students will expand their revision task schema and begin to internalize common surface-level errors, which will allow them to focus on amending more complex text-based changes during the revision phase of the writing process. The research study with *Revision Assistant, too* is being conducting with a group of eighth grade students at a school in the Pacific Northwest - the same school that inspired the design of the tool and provided secondary student perspectives of *Revision Assistant*.

ACKNOWLEDGMENTS

We would like to acknowledge Promevo for the development of *Revision Assistant*, with special thanks to Nathan Dixon and Aaron Gumz for their support during the development and updates.

REFERENCES

Beal, C. R. (1990). The development of text evaluation and revision skills. *Child Development, 61*(1), 247-258. https://doi.org/10.2307/1131063

Berzsenyi, C. A. (2001). Comments to comments: Teachers and students in written dialogue about critical revision. *Composition Studies/Freshman English News, 29*(2), 71-92.

De La Paz, S., Swanson, P., & Graham, S. (1998). The contribution of executive control to the revising of students with writing and learning difficulties. *Journal of Educational Psychology, 90*(3), 448-460. https://doi.org/10.1037/0022-0663.90.3.448

Faigley, L., & Witte, S. (1981). Analyzing revision. *College Composition and Communication, 32*(4), 400-414. https://doi.org/10.2307/356602

Ferretti, R. P., & Lewis, W. E. (2013). Best practices in teaching argumentative writing. In S. Graham, C. A. MacArthur, & J. Fitzgerald (Eds.), *Best practices in writing instruction* (2nd ed., pp. 113-140). New York, NY: The Guilford Press.

Hayes, J. R. (1996). A new framework for understanding cognition and affect in writing. In C. M. Levy & S. Ransdell (Eds.), *The science of writing: Theories, methods, individual differences, and applications* (pp. 1-27). Mahwah, NJ: Lawrence Erlbaum Associates.

Limpo, T., Alves, R. A., & Fidalgo, R. (2014). Children's high-level writing skills: Development of planning and revising and their contribution to writing quality. *British Journal of Educational Psychology, 82*(2), 177-193. https://doi.org/10.1111/bjep.12020

MacArthur, C. A. (2012). Evaluation and revision. In V. W. Berninger (Ed.), *Past, present, and future contributions of cognitive writing research to cognitive psychology* (pp. 461-483). New York, NY: Psychology Press.

MacArthur, C. A. (2013). Best practices in teaching evaluation and revision. In S. Graham, C. A. MacArthur, & J. Fitzgerald (Eds.), *Best practices in writing instruction* (2nd ed., pp. 215- 237). New York, NY: The Guilford Press.

MacArthur, C. A. (2016). Instruction in evaluation and revision. In C. A. MacArthur, S. Graham, & J. Fitzgerald (Eds.), *Handbook of writing research* (2nd ed., pp. 272-287). New York, NY: The Guilford Press.

McCutchen, D. (2006). Cognitive factors in the development of children's writing. In C. A. MacArthur, S. Graham, & J. Fitzgerald (Eds.), *Handbook of writing research* (pp. 115-130). New York, NY: The Guilford Press.

National Center for Education Statistics, & Educational Testing. (2012). The Nation's Report Card: Writing 2011. National Assessment of Educational Progress at Grades 8 and 12. NCES 2012-470. *National Center for Education Statistics.*

Neumann, K. L., & Kopcha, T. J. (*accepted for publication*). Using Google Docs for peer-then-teacher review on middle school students' writing. *Computers and Composition.*

Promevo. (2015). Revision Assistant [software]. Retrieved from https://chrome.google.com/webstore/d etail/revision-assistant/aalcpknmagohgaakdjhapkggbc aidepa?utm_source=permalink

Rijlaarsdam, G., Couzijn, M., & van den Bergh, H. (2004). The study of revision as a process and as a learning-to-write process: Two prospective research agendas. In L. Allal, L. Chanquoy, & P. Largy (Eds.), *Revision: Cognitive and instructional processes* (pp. 190- 208). New York, NY: Springer.

Scardamalia, M., & Bereiter, C. (1986). Research on written composition. In C. Wittrock (Ed.), *Handbook of research on teaching* (3rd ed., pp. 778-803). New York, NY: Macmillan.

Verheijen, L. (2013). The effects of text messaging and instant messaging on literacy. *English Studies, 94*(5), 582-602. https://doi.org/10.1080/0013838X.2013.795737