Improving Design Reviews at Google

Celal Ziftci
Google
New York, USA
celal@google.com

Ben Greenberg
Google
Vermont, USA
greenbben@google.com

2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE) | 979-8-3503-2996-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASE56229.2023.00066

***Abstract*—Design review is an important initial phase of the software development life-cycle where stakeholders gain and discuss early insights into the design's viability, discover potentially costly mistakes, and identify inconsistencies and inadequacies. For improved development velocity, it is important that design owners get their designs approved as quickly as possible.**

In this paper, we discuss how engineering design reviews are typically conducted at Google, and propose a novel, structured, automated solution to improve design review velocity. Based on data collected on 141,652 approved documents authored by 41,030 users over four years, we show that our proposed solution decreases median time-to-approval by 25%, and provides further gains when used consistently. We also provide qualitative data to demonstrate our solution's success, discuss factors that impact design review latency, propose strategies to tackle them, and share lessons learned from the usage of our solution.

***Index Terms*—design, design review, review and evaluation, peer reviewing, architecture review, engineering design**

I. Introduction

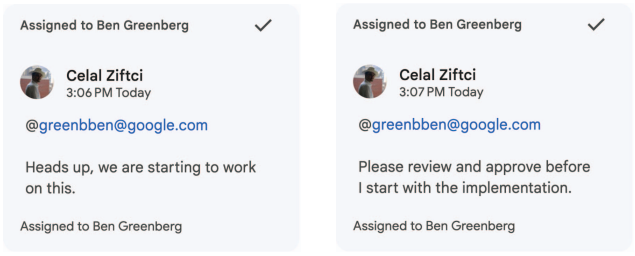
Design review is a critical and early stage in the software development process where stakeholders can provide design feedback, identify potential problems, and avoid costly mistakes in the subsequent steps of development [1]–[3].

Design reviews are widely adopted in the industry with more than 70% of the participants of a survey reported producing a requirements document or design [4]. Design reviews are also widely used across Google, where design documents are typically written in Google Docs [5] and stakeholders, including approvers, are added to the document by mentioning their emails in Google Docs comments or action items [6]. This design review process is neither structured nor automated, and has several shortcomings that hinder design review velocity: authors and approvers cannot track their design reviews easily, lack of actionable reminders lengthen the design review duration, commitments & approvals on the design are not recorded automatically.

In this paper, we propose a minimally invasive and generalizable technique to structure and automate the design review process through an integrated ecosystem of developer tools, present data based on the use of this technique over four years across Google, demonstrate how our technique dramatically improved design reviews for thousands of engineers, and summarize the lessons learned in the process.

II. Design Reviews at Google

Design reviews are widely used across Google. Although there are various tools available, Google Docs [5] is the most widely used tool to author engineering design documents.



(a) (b)

Fig. 1: Stakeholder added to a design document in Google Docs. The person in (a) is added for awareness, while the person in (b) is an approver. Approvers approve documents by resolving the action item assigned to them using the check mark at the top right corner.

After authoring, the owner of the design document typically adds stakeholders to the document by mentioning them with their email addresses inside Google Docs comments or action items [6], shown in Fig.1.

Distinguished using the comment's descriptive text, some people are mentioned for visibility and awareness, as in Fig.1(a), while others are *approvers* whose approval is required before the proposed design is implemented, as in Fig.1(b).

When a person is mentioned in a comment, they get an email in their inbox. Then, they typically add comments on the document for details, clarifications, and changes from the author, and after some back and forth discussions, they finally indicate their agreement or approval with the document by *resolving* the action item assigned to themselves [6] using the check mark at the top right corner shown in Fig.1.

Engineering design reviews are typically fluid, i.e. during discussions with approvers, authors may update their documents as needed even after certain approvers already approved the document, and there may still be unresolved comments on the document even after all approvals are obtained.

There are several shortcomings with this workflow from both the author and the approver perspectives.

First, relying only on the the emails sent to approvers is not ideal, as they have no distinguishing properties from the other emails, making it hard for approvers to keep track of design documents that need their attention, lengthening the design review unnecessarily.

DOI 10.1109/ASE56229.2023.00066

Fig. 2: Approver added to a design document in Google Docs using `#approver` in the descriptive text.

Second, since the author creates a separate comment per approver, she needs to check their respective comment manually when she wants to see the status of an approver.

Third, when a person is mentioned in a comment, they need to read and remember the comment's text to understand if they are an approver, which puts a mental burden on approvers.

Finally, it is commonplace for an approver to approve a document, yet add more comments for the author that do not impact their approval. As a result, it is not obvious to either the author or the approvers when a document is fully approved and ready to move on to the implementation phase, which creates additional friction for the authors to confirm with approvers using other communication channels such as internal chat.

As a result of these shortcomings, in a recurring Google survey called the Engineering Satisfaction Survey [7] that helps with understanding the needs of Google engineers, design review has been highlighted over several years as a significant pain point by both authors and approvers of design documents, and as a major hindrance to software development velocity, as full approval is typically desired and required to move on to the implementation phase.

III. Better Design Reviews With DAC

To bridge the design review shortcomings discussed in the previous section, we built a product named **D**esign **A**pproval **C**ompanion, DAC in short, with several components.

A. Approvers Table With DAC Google Docs Add-On

As developers typically use Google Docs to write design docs, we implemented an internal DAC Google Docs add-on [8] that allows design document authors to formalize their design reviews as a process with minimal additional effort.

An author installs the DAC add-on once, writes design documents as usual, and adds approvers using Google Docs comments as usual, with a small difference where she indicates the approver explicitly using the text `#approver` anywhere in the comment text, as in Fig.2. Approvers approve the document as usual, by resolving the assigned action item.

The DAC add-on periodically scans all comments in the document, identifies approvers using the `#approver` tag, generates an approvers table at the top of the document with status indicating text and colors, shown in Fig.3, and always keeps this table up to date automatically.

Approver	Status	Last Change
celal	APPROVED	2023-06-01
greenbben	PENDING	2023-05-23

Table auto-updated. Do not edit this section.

Fig. 3: Auto-generated and auto-updated approvers table placed at the top of the document created from Google Docs comments containing `#approver` in their descriptive text and showing the status of each approver.

This table addresses several of the shortcomings discussed in the previous section: authors can see pending approvers, approvers can see if the author is waiting on them, all readers can see the latest approval state of the document.

Additionally, instead of a new tool, both the authors and the approvers still use the same medium they are used to, i.e. Google Docs, which contributes to the ease of understanding and adopting the process.

Finally, the DAC add-on is generalizable, both for users of Google Docs, and for similar products that allow user interactions with comments in the industry.

B. Reminders With DAC Chrome Notifications

All Google developers have a default installed Chrome extension [9], called Event Notifier, that notifies them of various events that need their attention, e.g. code reviews and bugs, and keeps these events as an easily accessible list.

We integrated DAC into Event Notifier to notify authors and approvers, shown in Fig.4, (a) when a person is requested to approve a document, (b) when an approver approves a document, (c) when all approvers approve a document.

These notifications address several of the shortcomings discussed in the previous section: they keep authors and approvers engaged, they provide a central location to track incoming and outgoing design reviews for an individual, and they prevent design reviews from being forgotten, contributing to improved development velocity.

Additionally, these notifications are easily generalizable as extensions, similar to Chrome extensions [9], are publicly supported by many browsers.

C. Documentation Embedded DAC Team Review Tracker

At Google, teams typically keep track of their design documents, review them in weekly meetings to understand their status [4], whom they are waiting on, and whether there are any blockers that need to be addressed. Keeping the list and status of design documents up to date usually requires manual effort from the team members.

For documentation, Google has an internal tool named g3doc [10, Chapter 10] [11] that is used by teams for a wide

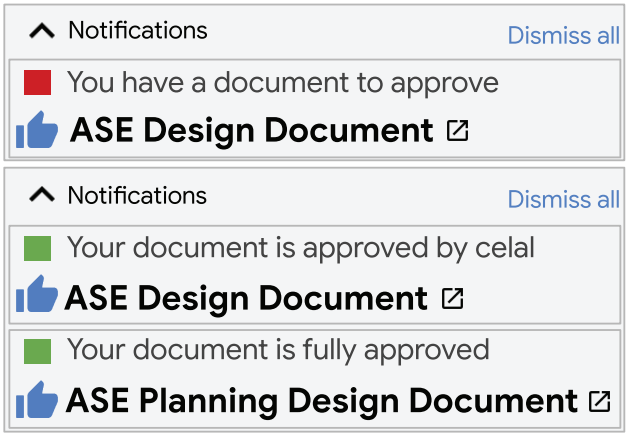


Fig. 4: Chrome notifications sent to approvers and authors on important design review related events.

Title	Author	Status
ASE Design Document	celal	PENDING: greenbben
ASE Planning Design Document	greenbben	APPROVED

Fig. 5: g3doc widget to automatically populate and display a list of design documents authored by a team's members and their status alongside their team documentation.

variety of needs, e.g. user documentation, internal documentation, system architecture summaries. g3doc supports building widgets, i.e. tools that can be embedded in documentation pages to display live data from other Google systems.

We built a DAC g3doc widget to automatically populate and display a list of all design documents authored by a team's members with their status, shown in Fig.5, so teams can avoid manual work during team reviews.

IV. Evaluation & Lessons Learned

In this section, we provide statistics on the usage of the different components of DAC at the time of writing of this paper (2023 June), the improvements it provided on design review velocity over the four years it has been used across Google, and lessons learned based on data and surveys.

A. Usability & Adoption

Table I summarizes the usage of the DAC components. 61,716 distinct users installed the DAC Google Docs add-on, 99,942 distinct users received at least one Event Notifier notification from DAC, and 115 distinct teams have been using the DAC g3doc widget to track their design reviews.

Table II summarizes the DAC add-on usage stats: 41,030 distinct authors used the DAC add-on on 251,517 design documents, and 141,652 of those documents are fully approved by all approvers. The authors are from 279 distinct Google offices across 20 different time zones in 48 countries. There are 74,796 distinct approvers added to at least one design document, and approvers are from 294 distinct Google offices

TABLE I: Statistics on the DAC components

# Distinct Google Docs add-on installs	61,716
# Distinct Event Notifier users that received a notification	99,942
# Distinct teams using the g3doc widget	115

TABLE II: Statistics on the DAC Google Docs Add-on usage

# Distinct authors	41,030
# Design documents	251,517
# Design documents approved	141,652
# Distinct author offices	279
# Distinct author countries	48
# Distinct author time zones	20
# Distinct approvers	74,796
# Distinct approver offices	294
# Distinct approver countries	52
# Distinct approver time zones	20
Median # approvers per document	2
Average # approvers per document	2.6

across 20 distinct time zones in 52 countries. There are a median of 2 and an average of 2.6 approvers per document.

DAC has several important usability features to prevent friction for authors and approvers: (a) DAC does not require extra actions from the author or the approvers other than using the `#approver` tag in Google Docs comments, (b) only the document author needs to install the DAC Google Docs add-on, there are no additional actions for approvers, and they typically do not even know that the approvals are handled by an add-on, (c) DAC is integrated into the users' critical workflows, and does not necessitate learning any new external tools or processes, (d) everything in the design review process except the required human interactions is fully automated.

Although we did not advertise DAC and there was no mandate for teams to use it, it has seen rapid adoption through word-of-mouth. Based on adoption rate and user surveys discussed in the following sections, engineers prefer a structured and automated design review process when available.

Lesson: Authors and approvers prefer and adopt a structured, automated design review process when available.

B. Baseline Case Study

Design reviews typically consist of several intertwined stages: authoring the design document, adding approvers, updating the document based on approver feedback and finally obtaining all approvals. The main purpose of DAC is to improve the design review velocity for engineers. To understand whether it helps serve this goal, we define time-to-approval as the metric of interest.

Time-to-approval (TTA): The total duration between the time the document was created and the time all approvers approved the document.

To serve as a baseline before creating DAC, summarized in Table III, we interviewed 44 distinct authors from 16 teams, and with their help, manually analyzed 108 design documents they authored that were fully approved without using DAC.

TABLE III: Statistics on the baseline case study

# Google teams surveyed	16
# Distinct design document authors from these teams	44
# Approved design documents from these authors	108
Median TTA in hours for these design documents	963

TABLE IV: Statistics on the documents that use DAC

# Approved design documents that use DAC	141,652
Median # documents by author that use DAC	2
Average # documents by author that use DAC	3.47
Median TTA in hours	722

We asked our interview participants to manually identify all comments in each document that constitute approvals of different approvers. We then calculated the duration between the timestamps of creation and the last approval comment of each document to find its TTA. Based on our analysis, the median TTA for these documents is 963 hours.

C. Using DAC Improves TTA

After the baseline case study, we implemented DAC and its components discussed in the previous section and collected stats on documents that used it over four years at Google.

Summarized in Table IV, 141,652 approved design documents used DAC, the median number of documents written by a distinct author is 2, the average number of documents written by a distinct author is 3.47, and the median TTA over the entire document population is 722 hours.

Compared to the baseline case study, the documents that use DAC have a 25% better TTA.

Lesson: Using an automated design review process that is well integrated into developer workflows meaningfully improves design review velocity.

D. Consistently Using DAC Improves TTA Further

We hypothesized that as users use DAC over time, their design reviews are likely to go faster due to a few reasons: (a) DAC clearly shows the status of both authors and approvers, (b) as users keep using DAC, the community likely understands the process better over time and they get more efficient, (c) DAC's workflow integrations lower friction and remind all stakeholders about their responsibilities continuously.

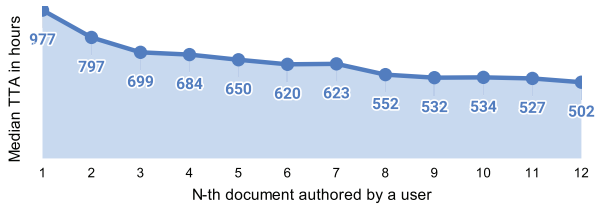


Fig. 6: Median TTA in hours for buckets of documents authored by users over time. Buckets are created chronologically, i.e. bucket 1 contains the 1^{st} document authored by each user, bucket N contains the N^{th} document authored by each user.

To assess our hypothesis, we analyzed the DAC document population by bucketing the documents authored by users chronologically, i.e. we bucket the 1^{st} document authored by each user, the 2^{nd} document authored by each user, the N^{th} document authored by each user. Using these buckets, we observed TTA on subsequent documents written by the same author, shown in Fig.6, where each bucket contains at least $1,000$ documents for significance.

First, compared to the TTA value of 963 hours for the baseline document population listed in Table III, users of DAC have a fairly close TTA of 977 hours on the 1^{st} document they use DAC. This is most likely because both authors and approvers are not yet familiar with DAC on their initial use.

TTA values on subsequent documents authored by the same user show significant improvements. Given that the average number of documents authored by a user is 3.47 listed in Table IV, the TTA value for $N = 4$ in Fig.6 is 684 hours. Compared to $N = 1$, this gives developers an average 29% improvement on getting design documents approved. As users continue using DAC, their TTA continues to improve further, supporting our hypothesis.

Lesson: As authors and approvers get used to the automation and integrations provided by a design review process, their TTA consistently improves over time.

E. DAC Provides Further Qualitative Benefits

In this section, we report the results of a survey we conducted with 7 senior managers whose organizations use DAC heavily, i.e. with at least 150 distinct document authors per organization, and a document per author ratio of at least 10. We asked open-ended questions to managers on why their organization uses DAC heavily, and what perceived benefits it provides, with some excerpts below.

We absolutely love the automation and workflow integrations of [DAC], it streamlined our org's design reviews.

I wrote a document for [...], and had it formally approved by several folks. It was great to have that formal approval on the document visible to everyone using [DAC].

Our work [...] inherently involves many teams and stakeholders, often in different locations and time zones. Lack of clarity around who is an approver and who has approved a document not only slows us down, but creates [..] frictions that can [...] cause conflicts and are hard to fix later. [DAC] [...] ensures that we move together as a cohesive, inclusive and psychologically-safe team.

In our org, [...] there are many projects that require long time commitments. We specifically use [DAC] to make sure those commitments are formally documented, approved and honored.

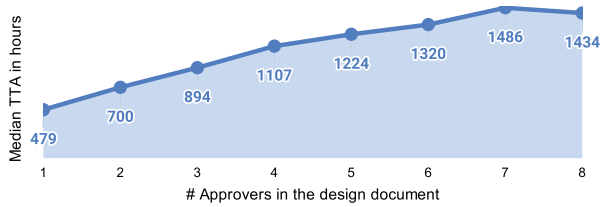


Fig. 7: Median TTA in hours by the number of approvers.

Based on these responses, DAC provides additional benefits for geographically distributed teams, for situations where commitments need to be documented and visible, and for cohesion that fosters psychological safety within teams.

Lesson: Automated design review processes benefit users by improving the review order, stakeholder agreement, stakeholder commitment and teams' psychological safety.

F. More Approvers Mean Longer TTA

Based on the baseline case study discussions and DAC surveys, engineers consistently stated that when there are more approvers, there is more discussion on the document, and more likelihood for some approvers to be in a different time zone or office, contributing to delays. In Fig.7, we plot TTA by the number of approvers on a document in the DAC population, and observe supporting evidence for this phenomenon.

Furthermore, engineers stated that they tend to prevent such delays by adding approvers slowly in waves instead of all at once, supported by the excerpts below.

> *We have a design review pipeline that [...] needs to strictly follow the order: Zurich, San Francisco, Sunnyvale, Zurich, and finally New York. [...] We use [DAC] to enforce this by adding approvers in that order.*
>
> ---
>
> *Adding all approvers at once is overwhelming. [DAC] helps us control this by slowly widening the list in waves.*

Lesson: More approvers positively correlate with longer TTA. One way authors try to avoid delays is by adding approvers to a design review in waves, starting from closer team members and progressively widening the list.

V. Threats to Validity

First, we built DAC for design documents, but we have no control over which documents users use it on.

Second, there are no guidelines on how to choose approvers and how many to have, it is completely decided by users.

Third, our baseline case study relies on manual human investigation where users may have made mistakes.

Fourth, we did not perform a controlled experiment between the users of our baseline study and the users of DAC, so our comparison may not be generalizable.

Finally, we do not control the life-cycle of projects, some projects may have been paused and picked up again at a later date, which might have introduced noise in TTA.

VI. Related Work

Design reviews provide benefits on early defect prediction and better software maintenance [12]–[15]. In line with the benefits of design reviews, exceptional engineers have been reported to spend more time in educating team members [16], in review meetings and consultations [17].

Structured walk-throughs [1] and inspections [2] are suggested as beneficial review mechanisms on different software artifacts. Software architecture review techniques have also been proposed to prevent defects early, including scenario based architecture analysis [18], architecture trade-off analysis method [19], active reviews for intermediate design [3], architecture level modifiability analysis [20], [21], and scenario based architecture re-engineering [22]. Recently, virtual reality has been suggested as a novel mode of interaction to support design reviews [23].

Design documents authored at Google typically cover various aspects of systems, including system views and scenarios [24], and architecture [25].

There are enterprise products for formal document approvals [26]–[28] where authors check-in their documents and ask reviewers for approval. Compared to the approach in this work, these products have two major shortcomings. First, they require users to use a separate product for approvals. Second, they typically require that no changes are made to the document during the approval stage, hence are not suitable for fluid engineering reviews that require documents to be unlocked for continuous edits during the approval stage.

VII. Conclusion

In this paper, we discuss design reviews, how they are conducted at Google, their shortcomings, propose an automated technique that meaningfully improves design reviews, and discuss lessons learned from its usage across Google over four years.

Based on our findings, developers prefer using an automated design review process, using such a process meaningfully improves design review velocity, consistently using this automation improves design review velocity further, while also helping individuals and teams to manage stakeholder agreements and commitments, and improving their psychological safety.

VIII. Acknowledgments

We thank the people that helped build DAC: Greg Dooley, Shondell Baijoo, Xavier Décoret, Trung Phan, Tony Zhao, Jason Lunn, Wentao Zhang, Maxim Goldin, Pedro Nunes, Vivek Parikh.

We also thank the internal Google reviewers, and the anonymous reviewers of the ASE committee.

REFERENCES

[1] E. Yourdon, *Structured walkthroughs*. Yourdon Press, 1989.

[2] M. E. Fagan, "Design and code inspection to reduce errors in program development," *IBM Systems Journal*, 1976.

[3] P. C. Clements, "Active reviews for intermediate designs," Carnegie Mellon University Software Engineering Institute, Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2000-TN-009, 2000.

[4] M. Ciolkowski, O. Laitenberger, and S. Biffl, "Software reviews, the state of the practice," *IEEE software*, vol. 20, no. 6, pp. 46–51, 2003.

[5] "Google docs: Online document editor," https://www.google.com/docs/about, 2023, [Online; Accessed: 2023-06-01].

[6] "Google docs: Use comments, action items, & emoji reactions," https://bit.ly/3q2rqeb, 2023, [Online; Accessed: 2023-06-01].

[7] L. Cheng, E. Murphy-Hill, M. Canning, C. Jaspan, C. Green, A. Knight, N. Zhang, and E. Kammer, "What improves developer productivity at google? code quality," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 1302–1313.

[8] "Google docs: Extending google docs with add-ons," https://bit.ly/3Ow7Iln, 2023, [Online; Accessed: 2023-06-01].

[9] "Chrome web store - extensions," https://bit.ly/3opZAYU, 2023, [Online; Accessed: 2023-06-01].

[10] T. Winters, T. Manshreck, and H. Wright, *Software engineering at google: Lessons learned from programming over time*. O'Reilly Media, 2020.

[11] R. MacNamara, "Do docs better: Practical tips on delivering value to your business through better documentation." USENIX Association, Jun. 2018.

[12] G. Abowd, L. Bass, P. Clements, R. Kazman, and L. Northrop, "Recommended best industrial practice for software architecture evaluation." Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, Tech. Rep., 1997.

[13] B. W. Boehm, "Verifying and validating software requirements and design specifications," *IEEE software*, vol. 1, no. 1, p. 75, 1984.

[14] B. Boehm and V. R. Basili, "Defect reduction top 10 list," *Computer*, vol. 34, no. 1, pp. 135–137, 2001.

[15] M. Fagan, "Design and code inspections to reduce errors in program development," *Software pioneers: contributions to software engineering*, pp. 575–607, 2002.

[16] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Communications of the ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[17] S. Sonnentag, "Excellent software professionals: Experience, work activities, and perception by peers," *Behaviour & Information Technology*, vol. 14, no. 5, pp. 289–299, 1995.

[18] R. Kazman, L. Bass, G. Abowd, and M. Webb, "Saam: A method for analyzing the properties of software architectures," in *Proceedings of 16th International Conference on Software Engineering*. IEEE, 1994, pp. 81–90.

[19] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere, "The architecture tradeoff analysis method," in *Proceedings. Fourth IEEE International Conference on Engineering of Complex Computer Systems (Cat. No. 98EX193)*. IEEE, 1998, pp. 68–78.

[20] P. Bengtsson and J. Bosch, "Architecture level prediction of software maintenance," in *Proceedings of the Third European Conference on Software Maintenance and Reengineering (Cat. No. PR00090)*. IEEE, 1999, pp. 139–147.

[21] P. Bengtsson, N. Lassing, J. Bosch, and H. van Vliet, "Architecture-level modifiability analysis (alma)," *Journal of Systems and Software*, vol. 69, no. 1-2, pp. 129–147, 2004.

[22] P. Bengtsson and J. Bosch, "Scenario-based software architecture reengineering," in *Proceedings. Fifth International Conference on Software Reuse (Cat. No. 98TB100203)*. IEEE, 1998, pp. 308–317.

[23] J. Wolfartsberger, "Analyzing the potential of virtual reality for engineering design review," *Automation in Construction*, vol. 104, pp. 27–37, 2019.

[24] C.-H. Lung, S. Bot, K. Kalaichelvan, and R. Kazman, "An approach to software architecture analysis for evolution and reusability," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*. Citeseer, 1997, p. 15.

[25] P. C. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford, "A practical method for documenting software architectures," 2002. [Online]. Available: http://repository.cmu.edu/compsci/671/

[26] "Get approvals on files in google drive," https://support.google.com/drive/answer/9387535, 2023, [Online; Accessed: 2023-06-01].

[27] "Create and test an approval workflow with power automate," https://learn.microsoft.com/en-us/power-automate/modern-approvals, 2023, [Online; Accessed: 2023-06-01].

[28] "Docusign," https://www.docusign.com, 2023, [Online; Accessed: 2023-06-01].