MOMENTUM

1. MOTIVATION

Programmers are notoriously terrible at estimating how long their projects will take to complete [1] [2] [3]. This is because developing software is inherently uncertain: it is impossible to anticipate every twist and turn of the design process at the outset. This is only exacerbated by the fact that many programming projects are not completed by a single person, but by teams — if an individual can't predict how long something will take for themselves, what hope does a team have of estimating the completion time for an entire project? Ultimately, this is a problem that can have huge monetary, emotional, and interpersonal costs [4]. The reality is that programmers are usually much too optimistic [1] [2] [3] — they try to fit too many features into a release, and some have to be cut; they think that they can focus effort in one area of a project because it will be quick, but it ends up taking two, three, four times as long as they had thought initially. Having better time estimates lets us choose how we focus our efforts [4] [5].

Despite the strong need for tools that make project estimation easier, existing effort estimation tools are either unintuitive (*cf. section 2.2*), require extensive knowledge of effort estimation processes (*cf. section 2.6*), or both. While learning these processes might be in a software developer's best interest, the reality is that many people (including software developers) are not good at making decisions that will benefit themselves long term [14]. Momentum aims to reduce the entry cost of accurate effort estimation by lowering the amount of time it takes to learn the tool, lowering the monetary cost of entry, and removing the need to have in-depth knowledge of effort estimation processes.

2. APPROACH

2.1. Goals

We are creating a tool that uses information gathered about a team of developers to predict a project's ship date. We leverage an existing, simple, and accurate software effort estimation strategy called Evidence-Based Scheduling (EBS) (*cf. section 2.6.2*) and are developing a new user interface for EBS that is easier and more accessible than existing solutions. We are developing a new user interface because existing tools for EBS either have poor visualizations of the data they produce, or simply lacks a front-facing app.

Our tool is called Momentum and is an open-source project available at https://github.com/cse403-MOMENTUM/MOMENTUM.

2.2. UI/UX User Studies: Comparing with Existing GUI Tools

Existing tools related to our project include Manuscript, ProcessPAIR, and Process Dashboard [9][10]. Manuscript is outlined below.

2.2.1. Manuscript

Manuscript is a web app for software teams that assists in the processes of project management, issue tracking, and support [8]. A subscription for a team of just five members

can cost from $62 to $75 per month. We decided to use this competing tool for our user research rather than other competing tools like ProcessPAIR and Process Dashboard for the following reasons:

1. Manuscript produces estimates for team-wide estimations, which aligns to the goals of our tool.
2. Manuscript uses EBS (*cf. section 2.6.2*), which is the effort estimation model we will use for the development of Momentum. This allows us to consider the effort estimation model a constant when performing usability comparisons between Manuscript and Momentum.
3. Manuscript's audience are software engineers who do not need prior knowledge of the Personal Software Process (PSP) model and processes, which most closely aligns to the audience that we want to build our tool for[6].

For the sake of transparency, we would like to point out that Manuscript includes more functions than just the uses related to EBS. We understand why this is done as Manuscript is intended to be a heavily-used tool, but we want our tool to be lightweight and have a sole purpose of providing teams with concrete estimations. Thus, we have only compared our tool to Manuscript's EBS functionality.

Manuscript has many functionalities that is not supported by Momentum because:

1. The functionality is not necessary in terms of helping the users estimate the ship date of their project
2. It is too time consuming to implement
3. We do not have the resources needed to implement the functionalities

In terms of tasks/cases, Manuscript has many data fields such as story points, backlog ordering, and kanban columns that we felt were unnecessary to include. There are also other Manuscript functionalities which are not related to task estimation. These functionalities include snippets, subscriptions, and webhooks. It also includes several ways to view milestones and tasks such as through an iteration planner and through a kanban board to help track cases in different ways. One of the features of Manuscript that we found to be interesting was Manuscript's integration with tools like Slack, GitHub, and Google Docs. Unfortunately, we currently do not have the time or the resources to add this feature to Momentum.

2.2.2. User Research

Because a large goal of our project is to make an extremely accessible tool, we are substantiating our accessibility decisions based on data collected from user research. Our first phase of this research was to see what users thought about our biggest competitor, Manuscript. The second phase of this research was to then to test the same users on completing the same tasks on Momentum 1.0 mockups. We then used the conclusions made from the comparisons between Manuscript and Momentum 1.0 to create Momentum 2.0. This process allowed our team to understand what users felt was difficult about using Manuscript,

avoid these problems in Momentum 1.0, and then fix all problems introduced in Momentum 1.0 in our final version, Momentum 2.0. We completed four informal user tests across Manuscript, Momentum 1.0, and Momentum 2.0. The users are detailed below and will be referred to as "User 1", "User 2", "User 3", and "User 4" for the rest of this portion of the report.

	Current Job/Student Status	Years on project development teams	Years spent programming	Major/ Degree
User 1	4th Year UW Student	1	6	CSE/HCDE
User 2	3rd Year UW Student	1	4	CSE
User 3	2nd Year UW Student	5	8	CSE
User 4	Network/Security Engineer	28	32	CSE/EE

The chosen tasks are summarized below as well as our intention behind choosing each task.

> Tasks 1-3: These tasks are presented as they represent a direct bridge between Manuscript and Momentum. Both tools should and do offer the ability to create a space to hold related tasks, create and modify tasks, and start, stop, and resolve these tasks. Because these tasks are shared by both Manuscript and our tool, we believe that we can directly measure our success in our usability improvements by comparing user experience in this task across both tools. For all tasks, the users were started on the "Main Iteration Planner" screen of Manuscript and the "Project View" screen of Momentum.
>
> Overarching Questions: In this section, several overarching questions are asked of the user to conclude and summarize the user's experience with the tasks given. We ask these questions to break away from the structure of a task and allow users to voice any other concerns or compliments of the tool after they have finished completing their task. Cross-referencing the responses from this section with the performance of users in each task allows us understand how much ease or complication in each task contributed to the user's overall view of the product.
>
> User 1 and User 2 completed all portions of User Testing on Manuscript before testing Momentum 1.0. User 3 and User 4 completed all portions of User Testing on Momentum 1.0 before testing Manuscript. This was done to lessen any bias that could occur by seeing one tool before another. All users completed User Testing of Momentum 2.0 last.

Task 1: Creating a new project

In this section, users were given the following scenario: "You are a part of a software project in which you have already broken down the tasks that are assigned to you and your fellow peers.

Create a place to store these tasks and fill out any forms to your best ability." The details of each user's interactions for each tool are detailed in the below tables.

Completion of Task as Directed:

	User 1	User 2	User 3	User 4
Manuscript	✗	✗	✓	✗
Momentum 1.0	✓	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Completion of Task in 90 seconds:

	User 1	User 2	User 3	User 4
Manuscript	✗	✗	✗	✗
Momentum 1.0	✓	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Supplementing Comments: All users except User 3 in the Manuscript task completed the task incorrectly. These Manuscript users also did not understand how to fill out more than half of the form options in the New Case screen. Manuscript User 3 took more than 90 seconds to locate the "New Project" button. All Momentum users (across both versions) found the "New Project" button in under 15 seconds.

Task 2: Creating a new task

In this section, users were given the following scenario: "Create a task and assign it to yourself, filling out any forms to your best ability." The details of each user's interactions for each tool are detailed in the below tables.

Completion of Task as Directed:

	User 1	User 2	User 3	User 4
Manuscript	✓	✓	✓	✓
Momentum 1.0	✓	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Completion of Task in 90 seconds:

	User 1	User 2	User 3	User 4
Manuscript	✗	✗	✗	✗
Momentum 1.0	✗	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Supplementing Comments: As in task 1, all Manuscript users did not understand how to fill out more than half of the New Case form options. All Manuscript users took longer than 60 seconds to create the case. Momentum 1.0 User 1 had trouble filling out the “Priority” field in the “New Task” form which caused the user to take longer than 60 seconds to fill out the form. Other Momentum 1.0 users also commented on this aspect of the form, but they still finished in under 60 seconds. Momentum 2.0 users said that they were happy that this the “Priority” field had been changed and had no trouble finishing the task.

Task 3: Altering created tasks

In this section, users were given the following scenario: “Enter into the task that you have previously created, indicate how you would begin working on the task, stop working on the task, and resolve the task.” The details of each user’s interactions for each tool are detailed in the below tables.

Completion of Task as Directed:

	User 1	User 2	User 3	User 4
Manuscript	✓	✓	✓	✓
Momentum 1.0	✓	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Completion of Task in 90 seconds:

	User 1	User 2	User 3	User 4
Manuscript	✗	✗	✓	✓
Momentum 1.0	✓	✓	✓	✓
Momentum 2.0	✓	✓	✓	✓

Supplementing Comments: Manuscript User 1 and User 2 did not finish within 60 seconds because they could not locate the case that they had previously created. Manuscript User 3 and User 4 also could not find their case due to the same reason, but searched for their case by using the chrome chrome “Find” search. Manuscript users commented that they didn’t

understand why they needed to exit the “Case View” screen and re-enter it to access the “Stop Work” button after they had pressed the “Start Work” button. Momentum users didn’t seem to have any trouble with this task. There were no negative comments about Momentum 1.0 and 2.0 by users concerning user interface design.

Conclusion: Overarching comments

In this section, users were asked questions about their experience using Manuscript. The questions are repeated below as are the answers from each user.

Question 1: Did you have any troubles using [Tool Name]? If so, what were they?

	User 1	User 2	User 3	User 4
Manuscript	Yes	Yes	Yes	Yes
Momentum 1.0	Yes	No	No	No
Momentum 2.0	No	No	No	No

Manuscript: Users 1, 2, and 3 mainly focused on their troubles with not being able to find the “New Project” button. All users also mentioned their inability to fill out most of the “New Case” form options.

Momentum 1.0: User 1 mentioned that they did not understand the “Priority” form option on the “New Case” screen.

Momentum 2.0: No comments from users.

Question 2: What did you like about [Tool Name]?

Manuscript: Users mentioned that they liked some of the “New Case” field options such as “Due Date” and “Priority”, that the Iteration Planner looks and acts like “Trello” (a web-based project management application), and that it shows complexity well. User 4 did not like anything about Manuscript so no answer was given for this question.

Momentum 1.0: All users responded that they liked how clean and intuitive Momentum was to use for the tasks. User 3 responded that the graph showing stats about project tasks could be helpful in a quick glance. Users 1 and 2 mentioned that they liked the progress bar included in the “Cases List View” Screen.

Momentum 2.0: Users gave very similar answers as were given above. Users also noticed the change in the “Priority” field of the New Task page as well as the new homepage and use of modals instead of new pages for task and project views. They commended all of these changes.

Question 3: What did you dislike about [Tool Name]?

Manuscript: Users repeated mentioned how “complicated” and “cluttered” the tool looked and felt while using it. The lack of explanations or small descriptions was also

mentioned more than once. User 1 mentioned that there seemed to be many "unnecessary bits" that were confusing.

Momentum 1.0: Users mentioned the confusion around the setup of the "Priority" field of the "New Case" form. User 3 suggested that the colors on the "Project View" screen could be less colorful. Nothing else was mentioned.

Momentum 2.0: User 3 mentioned that they would want the tool to handle more complexity. User 1 mentioned that they would have liked the mockup to have more interactivity (such as typing in text boxes) to experience the tool fully. Nothing else was mentioned.

Question 4: Would you use [Tool Name] for future projects if you needed to keep track of project-wide estimations?

	User 1	User 2	User 3	User 4
Manuscript	No	No	Maybe	No
Momentum 1.0	Yes	Yes	Maybe	Yes
Momentum 2.0	Yes	Yes	Maybe	Yes

Manuscript: User 3 mentioned that Manuscript would probably work for larger teams with complexity, but not for small teams. User 4 mentioned that "it might take me more time to figure Manuscript out than finishing my project".

Momentum 1.0/2.0: User 3 mentioned that Momentum (versions 1.0 and 2.0) would be good in a small team, but not a large team. No other comments were made.

Question 5: Would you recommend Manuscript to others?

	User 1	User 2	User 3	User 4
Manuscript	No	No	Maybe	No
Momentum 1.0	Yes	Yes	Yes	Yes
Momentum 2.0	Yes	Yes	Yes	Yes

Manuscript: User 3 said that the answer would depend on the use case, but definitely not for new teams.

Momentum 1.0/2.0: No additional comments provided by users.

Question 6: How would you rate your experience with [Tool Name] with 1 being the lowest and 5 being the highest?

	User 1	User 2	User 3	User 4
Manuscript	1.5	2.0	2.5	2.0
Momentum 1.0	4.5	4.0	4.5	4.0
Momentum 2.0	4.9	4.0	4.6	5.0

Manuscript and Momentum 1.0: No additional comments provided by users.

Momentum 2.0: User 2 gave the same rating as last time, mostly likely due to their indifference to the changes made between the two versions of Momentum: they did not comment on the “Priority” field, nor the Project View screen. All Momentum 2.0 users were recanted their scores for both Manuscript and Momentum 1.0 in order to scale their rating appropriately since their earlier user tests were given several weeks ago.

Concluding Remarks

Overall, users found Manuscript hard to use and were not always sure that they were completing the tasks correctly. They found that many fields and interfaces were not explained well enough for a first time user to fill out. The users would rather use other methods to estimate their projects than to use Manuscript, which leads us to believe that there is room for vast improvement in the usability of a tool that can help with project estimation. We strongly believe that a tool with a seamless user interface is a very novel contribution that drives users to begin using and continue using our tool.

Users found Momentum 1.0 much easier than Manuscript in the completion of their tasks. Users continually performed better in the same tasks in Momentum 1.0 than Manuscript. We then addressed the problems found in Momentum 1.0, such as the problems with filling out the “Priority” form option and address them in a new version of our tool. This new version of our tool, Momentum 2.0, was then tested to ensure that our changes had a positive impact with users.

2.3. Our Approach

We are designing Momentum to have a better user interface than Manuscript. Our primary goal is to create a new UI that lowers the barrier to entry for using software effort estimation methods. Our tool will be easy to learn so that users can quickly set up their projects and begin making estimations. Once a project is set up using Momentum, tracking times within the app takes only a few minutes each day.

We also improve the way EBS output is visualized. By leveraging recent research on uncertainty visualization from the UW’s Interactive Data Lab, we provide graphs that should allow our users to act on the data they receive better than Manuscript does for its users [11].

Because EBS emphasizes simplicity, it does not model task dependencies. Since we are not improving upon the EBS model, we also do not support task dependency modelling. We do, however, implement a priority system in Momentum. Users are encouraged to mark tasks that have many dependencies with a higher priority than other tasks to signal that those tasks should be completed first. Priorities do not affect the underlying model of effort estimation.

Below are specific examples of the improvements we have made to the GUI through two versions. We see our direct competitor as Manuscript, so we will compare our GUI mockups with Manuscript to display differences in design ideology.

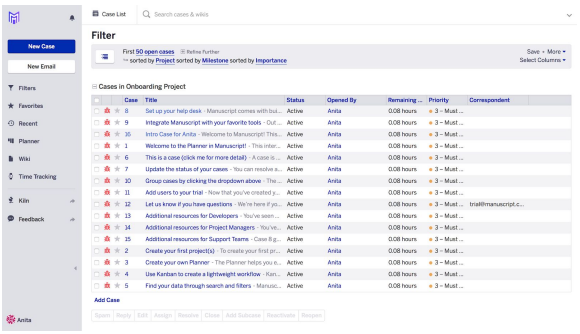

Fig.2.3.1.a.

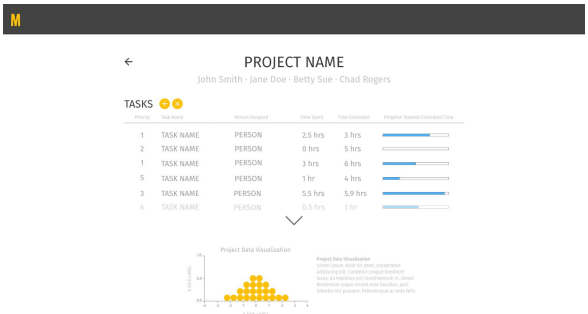

Fig. 2.3.1.b.

Fig. 2.3.1.a. is the task board for Manuscript while Fig. 2.3.1.b. is the task board for Momentum version 2.0. Momentum's task board is improved in that the interface is significantly simplified as our task board doesn't overwhelm the user with a multitude of non-important features in terms of making an estimate. Momentum's task board has data visualizations easily accessible on the page, which summarize the project's time projections and estimations. It shows tasks ranked by the order that the user chooses (such as ordering by newest tasks first, or ordering by deadlines). The user can also easily see the severity/priority of each task. An additional feature included in Monument is that the user can visually see the progress/time bar of each task.

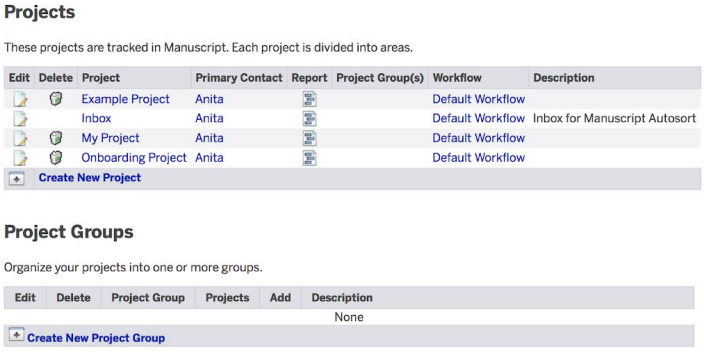

Fig. 2.3.2.a.

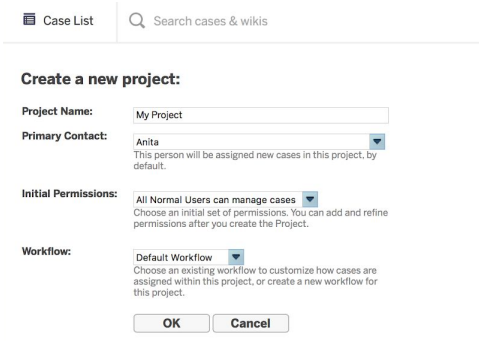

Fig. 2.3.2.b.

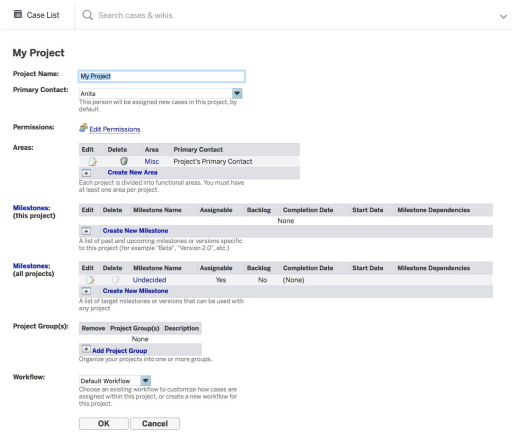

Fig. 2.3.2.c.

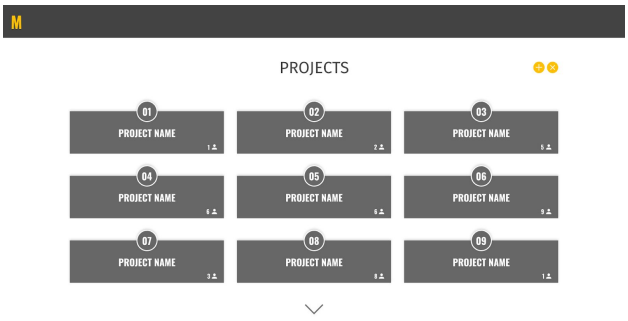

Fig. 2.3.2.d.

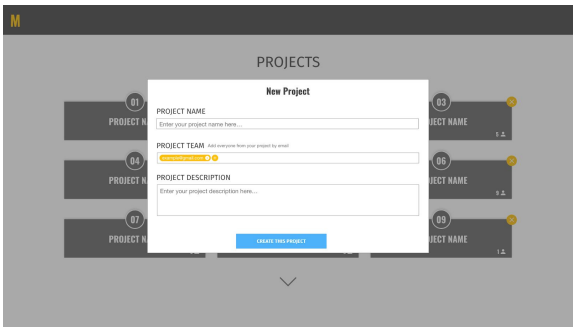

Fig. 2.3.2.e.

Another example where Manuscript lacks in user friendliness is that the user is required to click on a small link in the navigation menu in order to view their projects, and then require two separate screens (Fig. 2.3.2.b. And Fig. 2.3.2.c.) just to create a new project. Their screens (Fig. 2.3.2.a.) also provide too many options which require reading the manual to discern. For example, there is no clear indication on what the difference of Milestones vs Tasks are. There is also no explanation for the usage and differences of Areas vs Case Groups either. Momentum (Fig. 2.3.2.d.), on the other hand, avoids these confusions as its project page involves only one screen which elegantly displays all the user's projects after logging in, and then allows the user to add information about a new project without being redirected to a new page (Fig 2.3.2.e.).

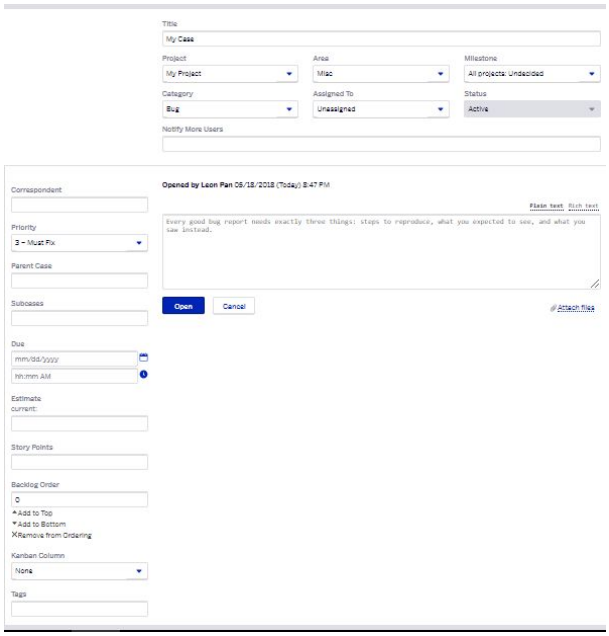

Fig. 2.3.3.a.

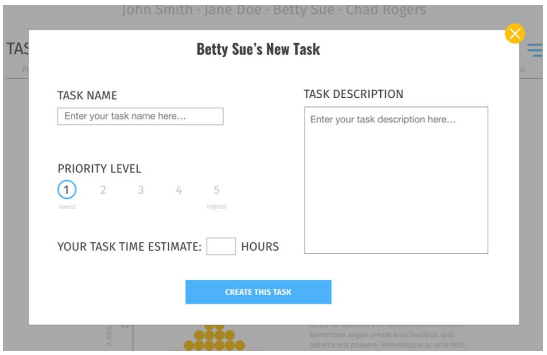

Fig. 2.3.3.b.

Fig. 2.3.3.a. is the new case page for Manuscript while Fig. 2.3.3.b. is the new task page for Momentum. Momentum's new task page is also significantly simplified as our task page doesn't overwhelm the user with an abundance of form options. Note that in the Manuscript new case page, most of the form options could be left black, including the estimate. Also, notice that the estimate field is below the button to open/create the case, meaning that the users might not even scroll far enough down to see the field before opening the case. This means the user might not input an estimation which is critical for EBS to calculate a projected ship date. Momentum's new task page only include the form options that are the most important in terms of knowing what the task is and making the estimate.

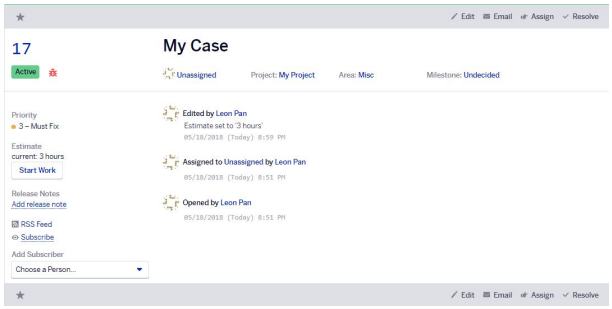

Fig. 2.3.4.a.

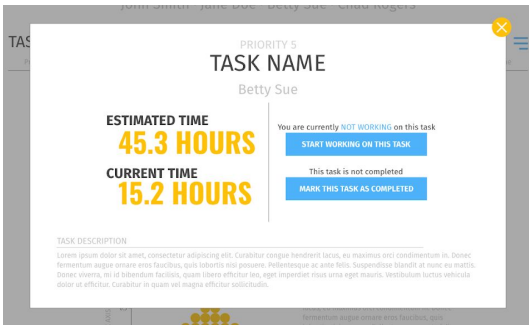

Fig. 2.3.4.b.

Fig. 2.3.4.a. is the case page for Manuscript while Fig. 2.3.4.b. is the task page for Momentum. Notice that in the Manuscript, the time currently spent on the case is not displayed anywhere on the page, so the user can not know how long he/she have been working on the case or compare it with the estimated time. Also note that in Manuscript, when the user click the start work button, he/she cannot stop the task without exiting and reentering the screen. Similarly, once the user clicks the stop work button, he/she cannot stop the task without exiting and reentering the screen. Momentum's task page provides the user with the information he/she might need to keep track of his/her progress and allows the user to easily manage the task from the task page.

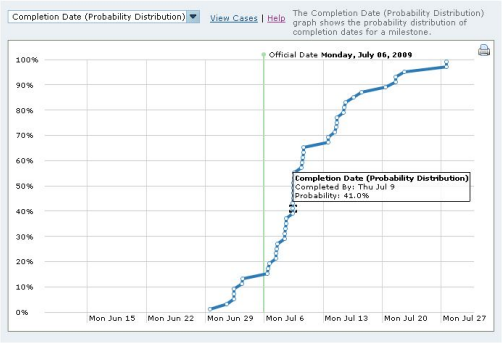

Fig. 2.3.5.a.

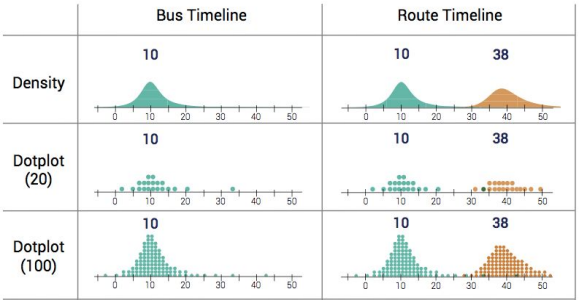

Fig. 2.3.5.b.

Uncertainty visualization in Manuscript vs. Momentum (examples from [11]).

Rather than visualizing ship dates as a line representing the cumulative probability of project completion, we propose visualizing times as a quantile dotplot. Researchers have found [11] that users are able to better interpret the uncertainty of a predicted time when visualized as a quantile dotplot than as a cumulative distribution.

> The user manual for our tool may be viewed at this link:
> https://github.com/cse403-MOMENTUM/MOMENTUM/blob/master/user-manual.md

2.4. GUI Mockup

Our GUI Mockup has progressed through two versions. The first version was based off of comments made about Manuscript during user testing, both concerning what users liked and disliked about the tool. The second version was based off of comments made about the first version of Momentum during user testing. A limited interactive experience with each version is available below:

Version one of Momentum:

> https://xd.adobe.com/view/b2653ac4-d8e2-4d64-90e0-d5cb276f5fb3/

Version two of Momentum:

> https://xd.adobe.com/view/81870323-070d-4e8a-97b6-b12b52dd486d/

2.5. Architecture

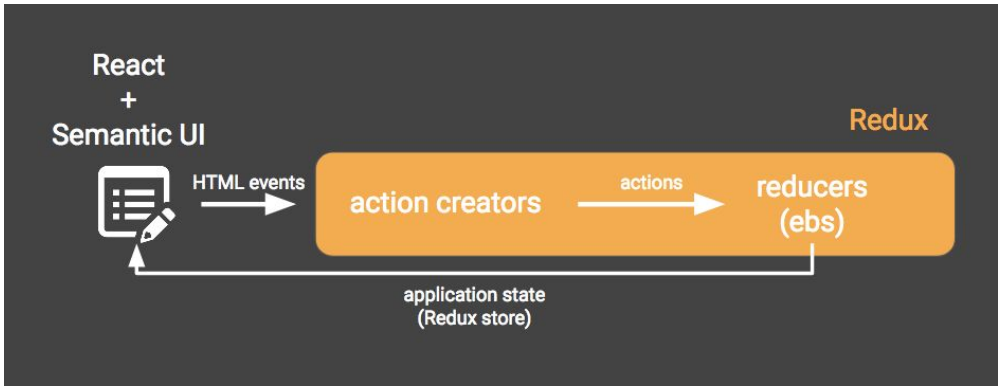

Fig. 2.5.1 Architecture

Our app is divided into a frontend and backend, written almost entirely in Typescript. We use several libraries and tools to manage the complexity of Momentum. For their software engineering benefits we use Typescript (a typed superset of Javascript), Jest (a Javascript testing framework), TSLint (a Typescript linter), and ImmutableJS (a collection of immutable data structures). These allow us to easily write tests, use rich types, and avoid unwanted state mutation. For the frontend we use React to define the layout of our web pages and SemanticUI to describe how the look and feel of our user interface components. For the backend, we employ Redux, a library for handling application state that works well with React and SemanticUI. Many of these the user interface components only store ephemeral data and handle extremely small computations (e.g. dispatching actions in response to events). Instead of storing and manipulating data themselves, these components send "actions" (*cf. Redux documentation*) to our Redux backend in response to events initiated in React components, and then receive an updated state from Redux, after Redux has performed this given action. EBS is implemented within Redux and forms the core business logic (Fig. 2.5.1).

3. Results

3.1 User Research

An important feature of our tool is to tailor itself to the individual, therefore, it is best to conduct experiments that involve case studies with programmers. Here, we will be highlighting the most important insights gained from our user research across each tool.

The user research on Manuscript indicated that there was much room for substantial improvements in user experience. Most importantly, we saw that Manuscript has non-intuitive presentation and access to project, milestone, and case creation buttons as well as broadly confusing form instructions without proper explanations. Outside of actual usability, users found that the user interface was boring and cluttered.

The user research on Momentum 1.0 indicates that many of the previously mentioned user experience problems with Manuscript have been elevated. When asked for a experience score, users on average reported a score much higher than the one given to Manuscript; however, users still found some difficulties with our tool. Because our project is centered on usability, we wanted to fix these problems with user experience even though Momentum 1.0 had already showed a significantly more friendly user experience than Manuscript.

User research conducted on Momentum 2.0 confirmed that our changes in interface had fixed many specific problems that users had when using Momentum 1.0. Average user experience (when compared to that of Momentum 1.0) rose with every user except one, where that user's rating stayed the same. Users were very happy that we had listened to all of their user experience concerns and had responded to each accordingly. Seeing users respond so positively to having their concerns addressed really underlined the importance of user experience and cemented that our project being focused on user interaction design was a very novel contribution to any and all tools.

4. Related Works

4.1. EBS

Evidence-Based Scheduling (EBS) is a method and piece of software for effort estimation developed by Joel Spolsky at FogCreek and has been integrated into their product called Manuscript [7]. Like PROBE, EBS first begins with the team of developers breaking down a large software project into smaller subprojects. Individual developers then cut these up into tasks they think they can complete in no more than a day of work. Developers estimate the amount of time each component will take. The EBS software keeps track of how much time the component actually takes and creates a model over time of how good an estimator the developer is. The EBS software then uses this model to predict how long the rest of the project will take and produces a probability distribution for the estimated ship date of the project.

Although there are other estimation models such as PROBE [6] and COCOMO II [12] [13], we decided to use EBS for Momentum rather than other existing models like PROBE and COCOMO II for the following reasons:

1. EBS is much simpler to implement than PROBE and COCOMO II and still provide good estimates for the ship date forecast.
2. PROBE and COCOMO II inherently require more user interactions or a much more complex system.

5. Future Works

In the future, we can improve on Momentum in a few ways. One approach is to improve upon the EBS estimation model by incorporating elements from other existing effort estimation processes/models such as COCOMO II and PROBE. By incorporating these elements into Momentum, we could potentially provide the users with more accurate estimates of the ship date of their products.

We could also complete more user studies to obtain more feedback from our users. We would like to conduct studies on Manuscript and Momentum with small teams rather than just individuals. Most project that requires a ship/release date are developed by teams so it would be beneficial to know how useful the tools are for groups of people rather than just individuals. We would also like to distribute the Manuscript and Momentum to the users so that they could use the estimation tools for an entire project. This way, we could know if the tools were useful for the entire project or if the tool become an inconvenience for the users.

REFERENCES

[1] M. Jørgensen, “What We Do and Don’t Know about Software Development Effort Estimation,” *IEEE Software*, vol. 31, no. 2, pp. 37–40, Mar. 2014.

[2] M. Jørgensen, “The effect of the time unit on software development effort estimates,” in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2015, pp. 1–5.

[3] H. Barki, S. Rivard, and J. Talbot, “Toward an Assessment of Software Development Risk,” *Journal of Management Information Systems*, vol. 10, no. 2, pp. 203–225, Sep. 1993.

[4] I. Benbasat and I. Vessey, “Programmer and Analyst Time/Cost Estimation,” *MIS Quarterly*, vol. 4, no. 2, pp. 31–43, 1980.

[5] K. Srinivasan and D. Fisher, “Machine learning approaches to estimating software development effort,” *IEEE Transactions on Software Engineering*, vol. 21, no. 2, pp. 126–137, Feb. 1995.

[6] W. Humphrey, *The Personal Software Process (PSP)* (Technical Report CMU/SEI-2000-TR-022). Pittsburgh: Software Engineering Institute, Carnegie Mellon University. 2000.

[7] J. Spolsky, “Evidence Based Scheduling,” Oct. 2007

[8] “Manuscript - Project Management for Software Teams.” Manuscript - Project Management for Software Teams, www.manuscript.com/.

[9] “Introduction to ProcessPAIR.” ProcessPAIR, www.blogs.fe.up.pt/processpair/tutorials/intro/.

[10] “Functionality for Individuals.” The Software Process Dashboard Initiative, http://www.processdash.com/functionality-personal.

[11] Kay, Matthew, et al. “When(Ish) Is My Bus? User-Centered Visualizations of Uncertainty in Everyday, Mobile Predictive Systems.” UW Interactive Data Lab | Papers, ACM Human Factors in Computing Systems (CHI), 2016, idl.cs.washington.edu/papers/when-ish-is-my-bus/.

[12] “COCOMO II Model Definition Manual.” USC, http://csse.usc.edu/csse/research/cocomoii/cocomo2000.0/cii_modelman2000.0.pdf

[13] B. Boehm *et al.*, “COCOMO Suite Methodology and Evolution,” April 2005. http://csse.usc.edu/TECHRPTS/2005/usccse2005-509/usccse2005-509.pdf

[14] Hershfield, Hal. “Future self-continuity: How conceptions of the future self transform intertemporal choice. Annals of the New York Academy of Sciences.” 2011.

[15] J. Nielsen, "How Many Test Users in a Usability Study?", *Nielsen Norman Group*, 2018. [Online]. Available: https://www.nngroup.com/articles/how-many-test-users/. [Accessed: 27- Apr- 2018].

META

A. Hours Spent on Assignment:

```
    Leon Pan: 6 hours
  Haley Ruth: 9 hours
 Anita Leung: 6 hours
Josh Pollock: 6 hours
   Austin Ha: 6 hours
```