

## Product Manager Visualization Final Report

### Project Title:

The Product Manager Dashboard

### Students:

Becca Stanger and Todor Tzolov

### Project Goals:

---

To offer agile development Product Managers visualizations tools to monitor current project progress, analyze trends across past ones, and better plan for future ones.

In order to accomplish this goal, we wanted to offer tools for managers to accomplish the following tasks:

- Learn how their current release (project) timeframes would be impacted by removing or postponing product tasks or subtasks
- Examine individual departments and overall release team's past performance as a tool for predicting current performance
- Plan how postponed tasks or subtasks could be allocated to future releases

### Related Work:

---

Todor is writing a paper on a Product Management Toolkit for Bob Glushko's Info Systems and Service Design Course. This research has shed light on the excessive time spent by Product Managers on project management, necessitated by the challenges and frequent reprioritization in Agile Development environments that are often employed at software companies. Though various tools exist for tracking and project management, none of them offer good visualization of current, past or upcoming projects.

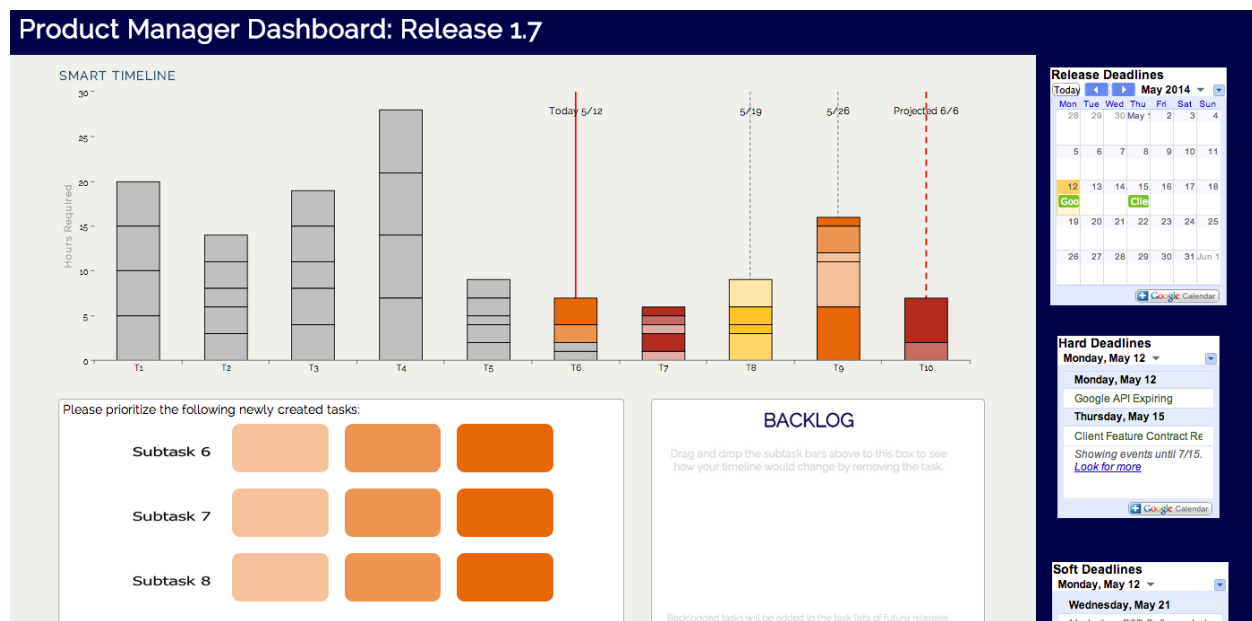
Becca has been working on a semester-long team project in Design and Development of Web Based Products and Services on calendar services. She had originally imagined designing a new student calendar system that allowed students to have a more holistic vision of time than Google Calendar offers. This goal turned out to be more challenging than she had imagined, but she has been thinking for awhile about different ways to visualize past, present, and future workloads, as this project did.

## Visualization:

We designed our product manager dashboard to focus on the three timeframes and tasks described in our goals - past, present, and future.

### The Present: Current Release Dashboard

The page the manager would automatically be directed to upon login would be the current product release homepage, which focuses on the present.



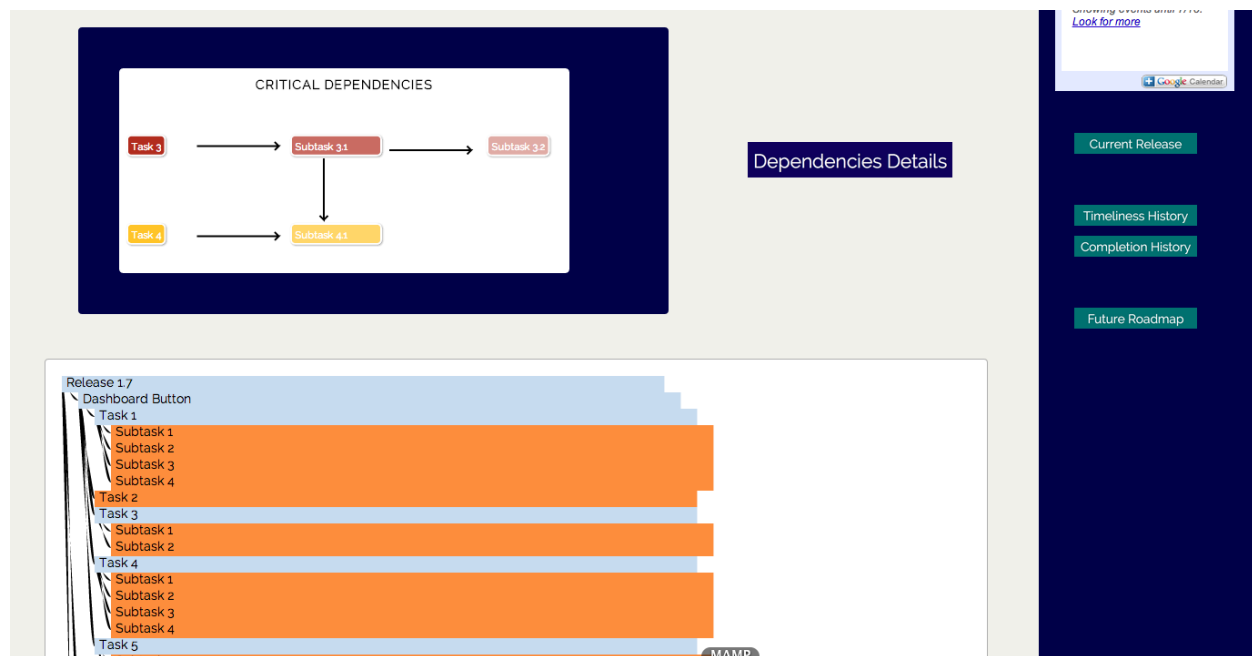
**Smart Timeline:** Immediately visible at the top is the smart timeline (see Appendix image A), which visualizes the overall progress of the current release project. Each task represents a key task, that is further broken down into subtasks. The tasks are color-coded by importance (red as the most important, orange as medium, and yellow as least important). The subtasks are also color-coded by importance with different shades of red, orange, or yellow. The most important red subtasks are bright red while the less important red subtasks are light red. The tasks that have already been completed are colored gray to remind the manager of the team's progress to date. Today's date is denoted by the solid red vertical line while the Projected Release date is labeled with the dashed red line at the end. The y axis is a measure of hours.

Given the color-coding and the y axis measure, a manager would be able to quickly assess which subtasks might be worth removing or keeping. For example, a solid red subtask with a small height would be a very important task that doesn't take very long, and thus

should be kept. Conversely a light yellow subtask that takes many hours should maybe be removed.

This timeline would be “smart” because it would automatically adjust the timeline based on a wide variety of factors, including the usual timeliness of the employees assigned to each task, reported unexpected delays, etc. It would also be smart in that it would automatically readjust when a manager tries to see the impact on the release timeline when a subtask is removed to the backlog.

**Backlog:** The backlog (see Appendix B) would theoretically be a box where managers could drag-and-drop a timeline subtask to in order to see how the timeline would adjust. The tasks added to the backlog would automatically be added to the Roadmap queue (more information to come).



**Dependency Flowchart:** When a subtask is dragged to the backlog, it would automatically generate a flowchart (see Appendix C) indicating what tasks and subtasks would be impacted by the removal of the subtask. The depiction of these relationships would allow managers to ensure they are not removing a task that could significantly impact others.

**Dependencies Details:** Clicking on the “Dependencies Detail” button would allow the manager to see an alternative view of how the selected subtask fits into the bigger picture (see Appendix D). The subtask that was in the backlog would be highlighted in the D3

Zoomable Pack visualization. The bubble sizes indicate how many hours a subtask might take. This visualization is added as an external link since our users had mixed reactions on whether they did or did not like that form of visualization.

**Collapsible Tree:** In addition to brushing and linking to the dependencies visualizations, a subtask added to the backlog would also be brushed and linked in the collapsible tree visualization (Appendix E). This visualization would offer one more alternative depiction of task hierarchy, but more importantly it would feature a sentence or two description of what the subtask is in case the manager wants to know more. The collapsible feature would allow the manager to control though how much information they see as to not overwhelm them.

**New Task Categorization:** In addition to the ability to see how the project timeline would change based on postponing tasks, the manager would also have a daily update box right below the timeline (Appendix F). This box would showcase the most recent new subtasks added by team members on the current tasks. Upon login, the manager would immediately determine the importance level of the subtasks by dragging and dropping them to the appropriate color shade. This new info would then be added to the timeline.

**Calendar Sidebar:** Finally, all pages would feature a calendar view with hard and soft company deadlines. The calendar would allow the manager to quickly understand how dates fit into a calendar understanding of time without pulling up her calendar. And the deadlines would be constantly featured to serve as a reminder of the deadlines that are easy to forget.

**Timebar (not shown):** We also originally had a timebar (Appendix G) that showed when the tasks were transitioning between the different teams (development, Q&A, integration, support). But at the presentation showcase on Friday we realized this visualization was not crucial to the audience and was easily overlooked. So we removed it for now until we come up with a better way to convey that info.

### **The Past: Timeliness and Completion**

**Timeliness History:** The timeliness page was intended to help managers understand how accurately his or her team has projected their hours in the past as well as how many hours they needed.

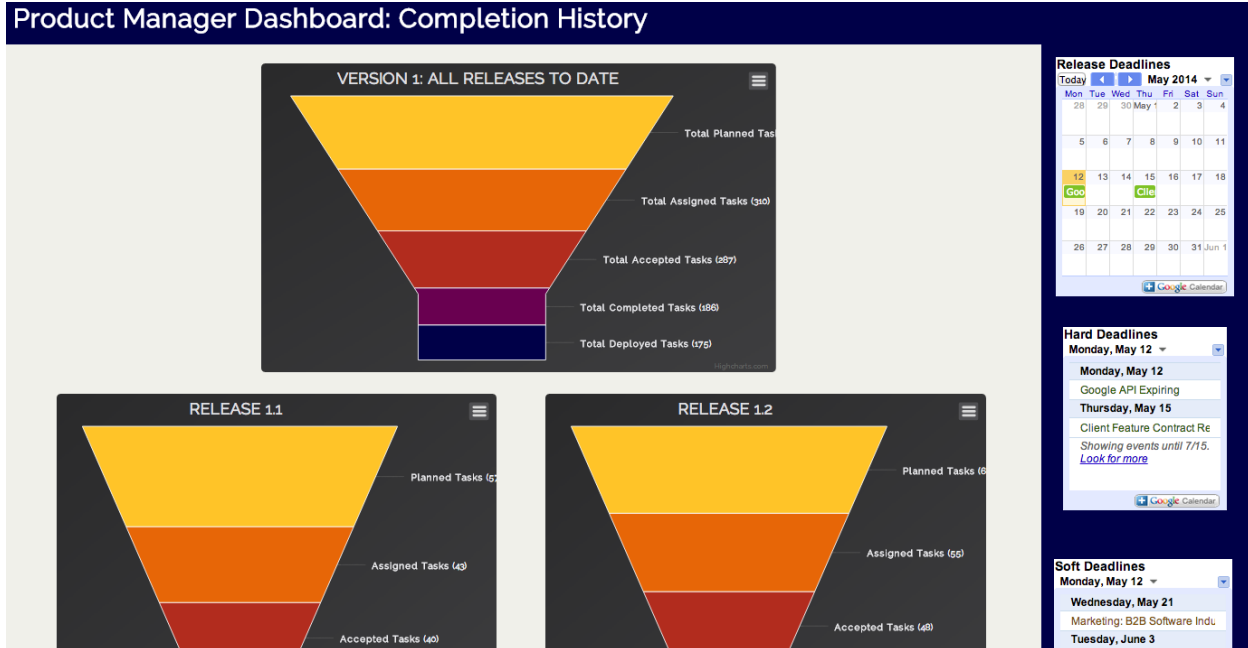
## Product Manager Dashboard: Timeliness History



One spider chart is used for each team (developers, q&a, integration, and support) - See Appendix H. One line expresses the estimated number of hours for each release of the product while the other expresses the actual number of hours spent on those releases. We considered simply depicting the difference between these two metrics to express time estimating accuracy but actually decided that the spider charts also allowed us to express how much time a team had spent, even if they estimated currently. Both accurate time estimates and minimal hours are important to a manager.

One person at Friday's showcase asked us why we didn't just use a horizontal timeline bar chart to express these numbers since the releases were chronological. But we felt the use of spider chart small multiples would allow managers to more quickly notice patterns and trends than a more linear bar chart.

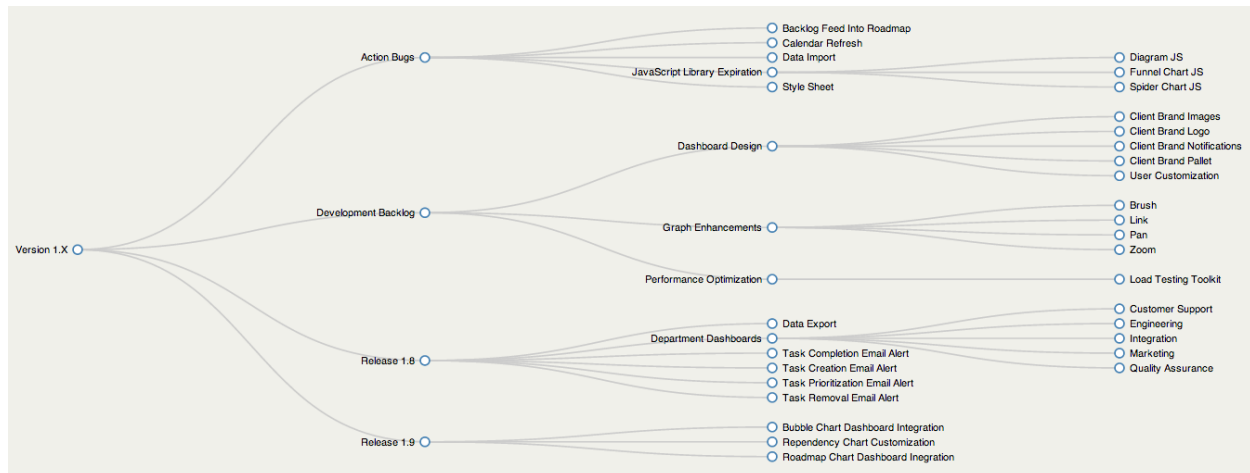
**Completion History:** Similar to the timeliness history page, the completion history page was intended to help managers understand how frequently his or her team has successfully implemented the tasks originally imagined. Only about 52% of imagined tasks are actually implemented in the Agile work environment. An increase in completion would increase productivity and progress.



We used funnel charts (see Appendix I) to convey, for each release, just how many tasks are actually funneled down from planned to assigned to accepted to completed to deployed. The yellow planned represents the total 100% of how many tasks are planned. The orange represents how many of those yellow task were actually assigned, etc. Ideally, a funnel with a large amount of dark blue would indicate that a large number of the planned tasks were successfully carried all the way to completion. As one viewer at the showcase pointed out though, we may want to reconsider this layout depending on the organization (see “Results” comments below).

### **The Future: Roadmap**

When coordinating a release, managers often will postpone tasks or subtasks if time is running out and hard deadlines are approaching. Its his responsibility, however, not to completely remove these subtasks but plan for them in future releases. To help managers plan for these future tasks, we created a drag-and-drop Roadmap.



Every time a subtask or task is moved to the backlog it would automatically autopopulate to the “Development Backlog” node pictured in the roadmap. In addition, whatever bugs developers would encounter while working through the product would be added to the “Action Bugs” node and queue.

When a manager has the time then to start planning for the future, she would be able drag-and-drop the tasks in these two queues to the planned future releases. Thus the manager would be able to have a rough idea of the component plans for future releases.

### Sum:

Together these pages are designed to offer managers an understanding of his team’s past, present, and future performance and respond accordingly.

### Data Used:

Since we only had two people on our team, we decided to not get caught up in wrestling with data beyond identifying its format and common field. We opted to focus more on how to design visualizations that fit our users’ needs. Instead, we hardcoded mock-up data into our designs with the assumption that if fully implemented, we could plug in data from sources like JIRA or Pivotal Tracker. Fortunately, this was confirmed by Armando Fox, Agile Development Professor at the I School, during his visit to our showcase.

Even though we did not use a dataset, we mocked up what a sample dataset we might need would look like.

	A	B	C	D	E	F	G	H	I
1	Unique ID	Release	Feature name	Task Order	Sub-task order	Critical (1-5)	Dev Complete	Q&A Complete	Deploy Complete
2	PD-1001	1.1	Dashboard Button	1	1.1	5	1	1	0
3	PD-1002	1.1	Dashboard Button	2	2.1	5	1	1	0
4	PD-1003	1.1	Dashboard Button	3	3.1	5	1	1	0
5	PD-1004	1.1	Dashboard Button	4	4.1	5	1	1	0
6	PD-1005	1.1	Dashboard Button	5	5.1	5	1	1	0
7	PD-1006	1.1	Dashboard Button	5	5.2	5	1	1	0
8	PD-1007	1.1	Dashboard Button	5	5.3	4	1	1	0
9	PD-1008	1.1	Dashboard Button	5	5.4	3	1	1	0
10	PD-1009	1.1	Dashboard Button	5	5.5	2	1	1	0
11	PD-1010	1.1	Dashboard Button	5	5.6	2	1	1	0
12	PD-1011	1.1	Dashboard Filter	6	6.1	5	1	0	0
13	PD-1012	1.1	Dashboard Filter	7	7.1	5	1	0	0
14	PD-1013	1.1	Dashboard Filter	8	8.1	4	1	0	0
15	PD-1014	1.1	Dashboard Filter	9	9.1	4	1	0	0
16	PD-1015	1.1	Dashboard Filter	10	10.1	5	1	0	0
17	PD-1016	1.1	Dashboard Filter	11	11.1	4	1	0	0
18	PD-1017	1.1	Dashboard Filter	11	11.2	4	1	0	0
19	PD-1018	1.1	Dashboard Filter	11	11.3	2	1	0	0
20	PD-1019	1.1	Dashboard Filter	12	12.1	5	1	0	0
21	PD-1020	1.1	Dashboard Filter	13	13.1	4	1	0	0
22	PD-1021	1.1	Dashboard Filter	14	14.1	4	1	0	0
23	PD-1022	1.1	Dashboard Filter	14	14.2	3	1	0	0
24	PD-1023	1.1	Dashboard Filter	14	14.3	2	1	0	0
25	PD-1024	1.1	Dashboard Filter	15	15.1	5	1	0	0
26	PD-1025	1.1	Dashboard Filter	15	15.2	5	1	0	0
27	PD-1026	1.1	Dashboard Filter	15	15.3	5	1	0	0
28	PD-1027	1.1	Dashboard Filter	15	15.4	5	1	0	0
29	PD-1028	1.1	Dashboard Filter	15	15.5	4	1	0	0
30	PD-1029	1.1	Dashboard Filter	15	15.6	4	1	0	0
31	PD-1030	1.1	Dashboard Filter	15	15.7	4	1	0	0
32	PD-1031	1.1	New visualizations	16	16.1	4	0	0	0
33	PD-1032	1.1	New visualizations	16	16.2	2	0	0	0
34	PD-1033	1.1	New visualizations	17	17.1	4	0	0	0
35	PD-1034	1.1	New visualizations	17	17.2	2	0	0	0
36	PD-1035	1.1	New visualizations	18	18.1	4	0	0	0
37	PD-1036	1.1	New visualizations	18	18.2	2	0	0	0
38	PD-1037	1.1	New visualizations	19	19.1	4	0	0	0
39	PD-1038	1.1	New visualizations	19	19.2	2	0	0	0
40	PD-1039	1.1	New visualizations	20	20.1	4	0	0	0
41	PD-1040	1.1	New visualizations	20	20.2	2	0	0	0
42	PD-1041								
43	PD-1042								
44	PD-1043								
45	PD-1044								
46	PD-1045								
47	PD-1046								
48	PD-1047								
49	PD-1048								
50	PD-1049								

## Tools Used:

We used a combination of D3, Highcharts, Powerpoint, Javascript, HTML, and CSS to create this dashboard.

### Highcharts:

- Timeline
- Dependency Flow Chart
- Funnel Charts
- Spider Graphs

### D3:

- Zoomable Pack Layout (bubbles)
- Collapsible Indented Tree
- Drag and Drop Collapsible Tree Layout



Javascript: Drag and drop new task prioritization box

Design: HTML, CSS, Powerpoint

### Steps Taken:

---

- Todor teaches Becca about Agile Development process
- Sketch out potential dashboard design based on our own assumptions and understanding
- Solicit feedback from classmates and product managers Todor knows professionally at two companies: Clarabridge and Marin Software
- Sketch out revised design based on feedback and new foci on being able to resort tasks
- Create visualizations in Highcharts and D3
- Create HTML and CSS framework for visualizations

### Results:

---

When Todor solicited feedback from actual product managers, it redirected the focus of our dashboard. Originally we had imagined that managers would like to know when tasks were going over or under the budgeted time and who possibly might be responsible for delayed tasks. After interviewing the product managers, however, we realized that this knowledge of why and who was a secondary concern to them.

More importantly, they wanted to focus on how they should redistribute tasks in their release timelines based on the changes. As a result, we moved the history pages on the team's completion and time estimating history to secondary pages. Instead, on the dashboard we showcased the timeline and the ability to prioritize tasks to see potential repercussions.

We also took Marti's advice about not just focusing on the problems a product manager faces, but also the accomplishments. In particular we intentionally kept completed tasks on the release timeline to help managers remember how far they have already come in the process and provide a sense of achievement.

We received quite positive feedback at the presentation showcase. Some possible changes we would make in the future based on that feedback include some slight style

changes to the visualization as well as a reconsideration of the funnel charts. As one person pointed out, having only some ideas funnel all the way down to deployment might actually be a natural process of ideation, depending on the company. If brainstorming is highly encouraged, there may be many more ideas offered than actually can be deployed and that may not be a problem.

Most notably, we receive good feedback from Armando Fox, who teaches an Agile Development course here and gave us guidelines on how we could link our interface to data from Pivotal Tracker's API. He confirmed that the data fields and terminology we use match those commonly found by software companies using Pivotal Tracker, which should a potential live data feed in the future an achievable goal. He confirmed that if made freely available our interface may gain widespread adoption as a Pivotal Plugin, since as a development tool built by developers, it currently badly lacks visualization interface.

### Links:

---

<http://people.ischool.berkeley.edu/~beccastanger/finalproject2/>

Subpages found through above link:

<http://people.ischool.berkeley.edu/~beccastanger/finalproject/bubbles/bubbles.html>

<http://people.ischool.berkeley.edu/~beccastanger/finalproject2/releaseshistory2.html>

<http://people.ischool.berkeley.edu/~beccastanger/finalproject2/funnels.html>

<http://people.ischool.berkeley.edu/~todor/projects/Tree/Tree/TreeRoadmap.html>

### Table:

---

Part	Becca	Todor
Wireframe Sketches	60%	40%
User interviews	-	100%
Data table mockup	-	100%
Timeline (Highcharts)	100%	-
Timebar (not shown, Highcharts)	100%	-

Prioritization Drag and Drop (Javascript)	-	100%
Static images (backlog, buttons)	100%	-
Dependencies Flowchart (Highcharts)	-	100%
D3 Zoomable Pack (D3 bubbles)	100%	-
Collapsible Tree (D3)	100%	-
Roadmap drag and drop (D3)	-	100%
Spider charts (Highcharts)	-	100%
Funnel charts (Highcharts)	-	100%
HTML / CSS	65%	35%

**Note:** Some of these tasks took more time and energy than others. Overall, we feel like we both put a lot of work into this project (especially with only two people on the team) and equally distributed the workload.

**Appendix:**

Image A - Smart Timeline:

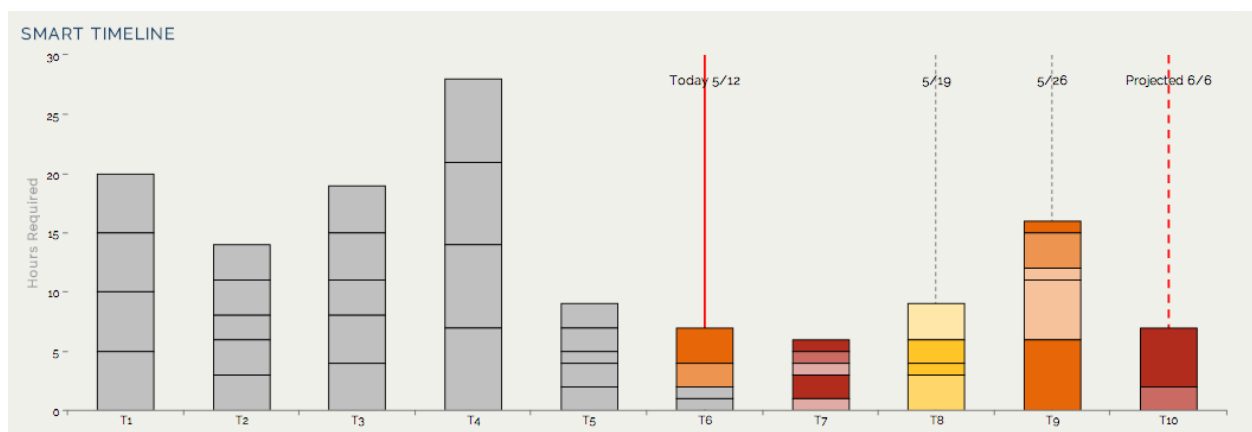


Image B - Backlog



Image C - Dependency Flowchart

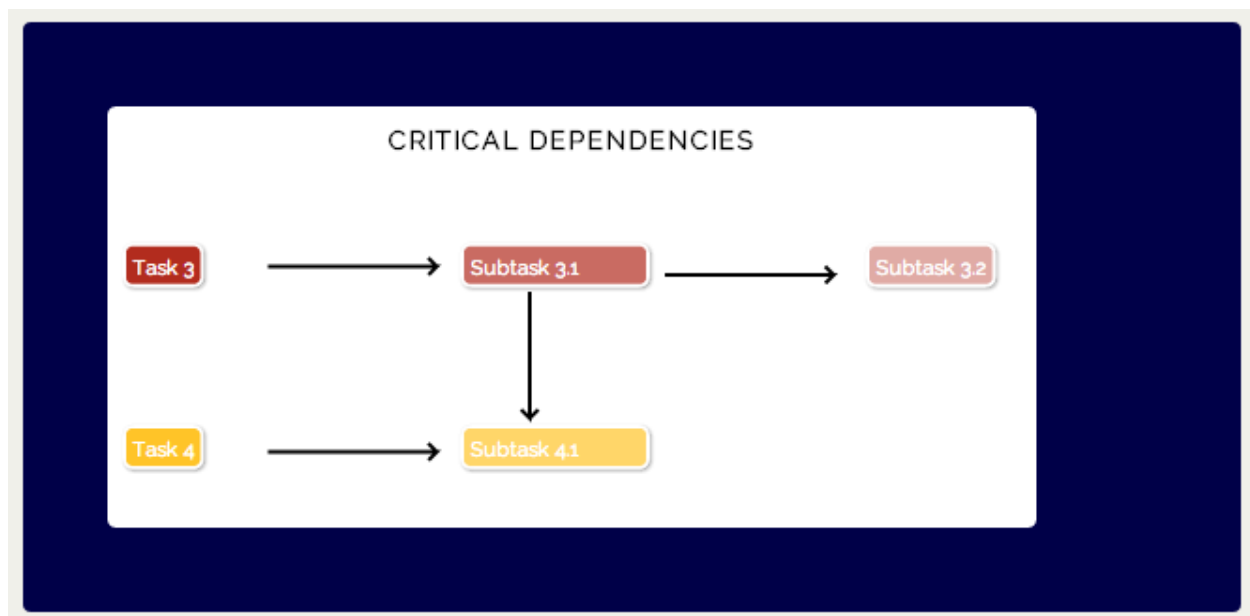


Image D - Dependencies Details

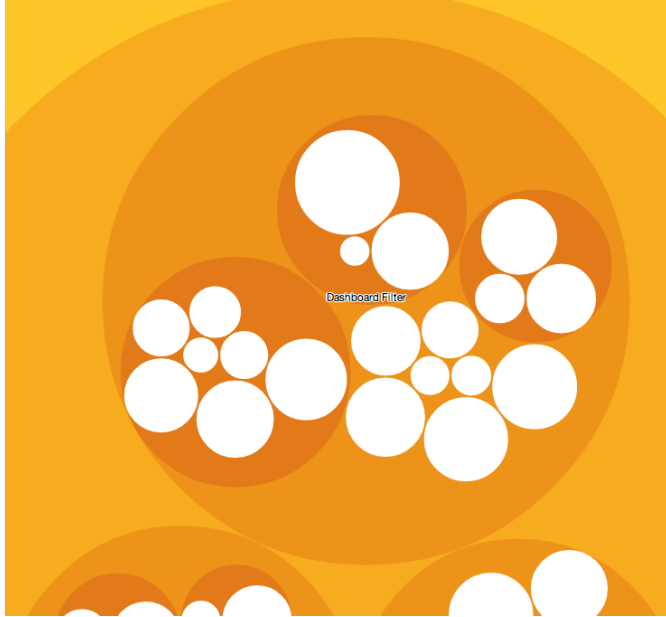


Image E - Collapsible Tree

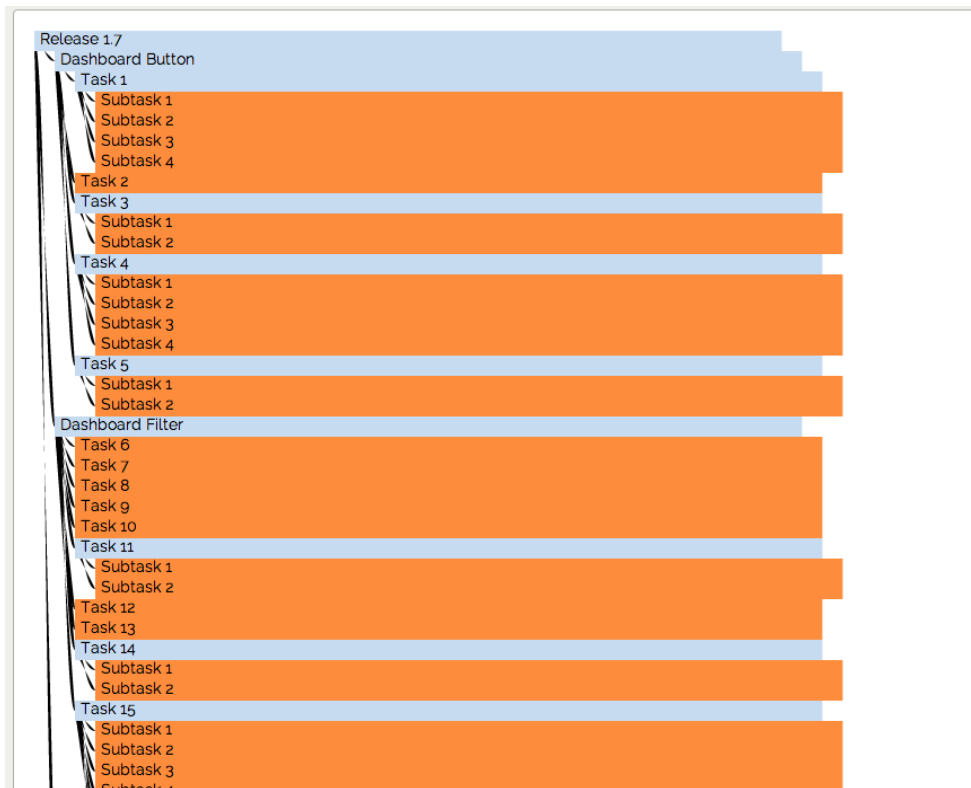


Image F - New Task Categorization

Please prioritize the following newly created tasks:

Subtask 6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subtask 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subtask 8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Image G - Timebar



Image H - Spider Chart

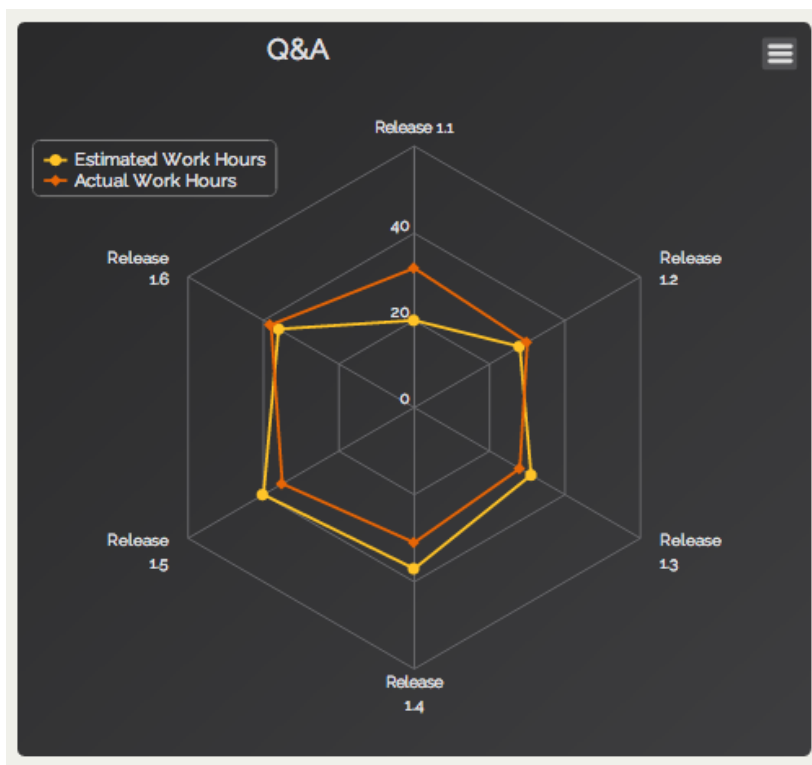


Image I - Funnel Chart

