

Software Project Planning, Design and Quality Management

High Distinction task 81HD



Dilni De silva

Student ID: 103616345

Tutorial: Tuesday 12:30 Hawthorn EN310

Background

Project: Pro. co

Company: Productivity_incopripiated

Software: A digital productivity application for scheduling and daily planning

Synopsis:

Productivity incorporated is a small start-up company that creates and implements consumer used productivity tools for professional and educational objectives. Productivity_incopripiated recently has come up with the concept of a digital planner desktop application. Which should include basic scheduling and task completion methods.

Application is brainstormed to have a to-do list where users can add a task, saving them, so when a user opens, the application again can retrieve see already entered task. These tasks should be able to be organised as well and be viewed in a daily, weekly, and monthly layout. Hence tasks written on certain dates are saved. In addition to this, the application must also have a notification system added to the scheduling function. Moreover, when all outlined criteria must be implemented with a GUI.

Backlog items

No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 2 3)
F1	Add task	-	10	Sprint 1
F2	Remove task	F1	10	Sprint 1
F3	Move task position in to-do list	F1	8	Sprint 1
F4	Add date scheduling for to-do list task	F1	7	Sprint 2
F5	Edit date scheduling for to-do list task	F1, F4	7	Sprint 2
F6	Add time scheduling for to-do list task	F1, F4, F5	7	Sprint 2
F7	Edit time scheduling for to-do list task	F1, F4, F5, F6	7	Sprint 2
F8	Create and implement view planner (where users can view their daily, weekly, monthly task scheduled from the to-do list)	F1-F7	9	Sprint 3
F9	Create and implement edit planner view (where users can edit their daily, weekly, and monthly task scheduled from the to-do list)	F1-F7	8	Sprint 3
F10	Create and implement a notification system (give notification options for daily, weekly, and monthly)	F1-F7	8	Sprint 3

Explanation for business value

Backlog item	Business Value	Explanation
F1	10	This backlog item implements the main feature of the desktop application. Hence has a higher business value. As well as other required features building on the functionality of this backlog item. Therefore this would be allocated a high business value.
F2	10	Also this backlog item is the main feature that has been asked from the client septically in the product background. In addition to this, the backlog item has a lot of dependencies.
F3	8	This is graded at an 8 due to this item being important as it is the main functionality and is a feature where the user can interact with things in the applications. However, hasn't been specified by the client as necessary hence a lower score.
F4	7	This item is Important as it enables the user to set time constraints on tasks however due to its functionality not being linked to other backlog items working, it shouldn't hold as much weight. Note that backlog items F4 - F7 have the same business value, hence the same explanation for the business value.
F5	7	This item is Important as it enables the user to set time constraints on tasks however due to its functionality not being linked to other backlog items working, it shouldn't hold as much weight. Note that backlog items F4 - F7 have the same business value, hence the same explanation for the business value.
F6	7	This item is Important as it enables the user to set time constraints on tasks however due to its functionality not being linked to other backlog items working, it shouldn't hold as much weight. Note that backlog items F4 - F7 have the same business value, hence the same explanation for the business value.
F7	7	This item is Important as it enables the user to set time constraints on tasks however due to its functionality not being linked to other backlog items working, it shouldn't hold as much weight. Note that backlog items F4 - F7 have the same business value, hence the same explanation for the business value.
F8	9	Backlog item is another feature that has been asked by the client to be implemented and outlined in the background of the project proposal. However, due to the lack of dependencies, it doesn't receive the 10 for business value. Note that the feature is similar to F9 and F10 so business value and explanation for business value are the same.
F9	9	Backlog item is another feature that has been asked by the client to be implemented and outlined in the background of the project proposal. However, due to the lack of dependencies, it doesn't receive the 10 for business value. Note that the feature is similar to F8 and F10 so business value and explanation for business value are the same.

F10	9	Backlog item is another feature that has been asked by the client to be implemented and outlined in the background of the project proposal. However, due to the lack of dependencies, it doesn't receive the 10 for business value. Note that the feature is similar to F8 and F9 so business value and explanation for business value are the same.
------------	---	--

Product Backlog item Schedule

Sprint 1 (16/05/2022 - 27/05/2022)

No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 2 3)
F1	Add task	-	10	Sprint 1
F2	Remove task	F1	10	Sprint 1
F3	Move task position in to-do list	F1	8	Sprint 1

Sprint 2 (30/05/2022 - 10/06/2022)

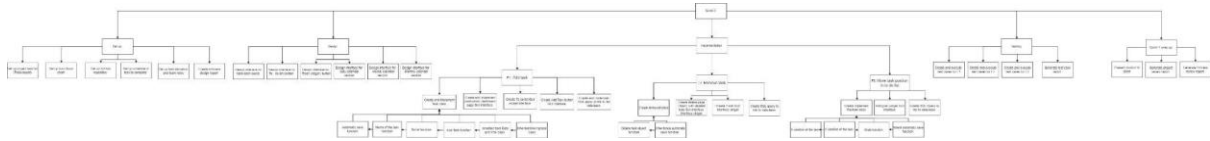
No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 2 3)
F4	Add date scheduling for to-do list task	F1	8	Sprint 2
F5	Edit date scheduling for to-do list task	F1, F4	7	Sprint 2
F6	Add time scheduling for to-do list task	F1, F4, F5	8	Sprint 2
F7	Edit time scheduling for to-do list task	F1, F4, F5, F6	7	Sprint 2

Sprint 3 (13/06/2022 - 24/06/2022)

No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 2 3)
F8	Create and implement view planner (daily, weekly, monthly where users can add new items from their to-do list)	F1-F7	9	Sprint 3
F9	Create and implement edit planner view (daily, weekly, monthly where users can add new items from their to-do list)	F1-F7	9	Sprint 3

F10	Create and implement a notification system (daily, weekly, monthly)	F1-F7	8	Sprint 3
------------	---	-------	---	-----------------

Sprint 1 plan Work breakdown structure



Note: Better and clever version of the [WBS](#)

Effort estimation Sprint 1

Equation

X = estimated time

$E = O + 4M + P / 6$

O = x minutes

M = 4 x M = x minutes

P = x minutes

Set up

Sub-task	Estimation of effort	Explanation
Set up project task list (Trello board)	<p>O = 30 minutes M = 4 x 60 = 240 minutes P = 120 minutes</p> <p>$E = 30 + 240 + 120 / 6$ $E = 170 / 6$ $E = 28.33 \text{ minutes} \rightarrow 30 \text{ minutes,}$</p>	<p>Optimistic: Estimated to be 30 minutes as setting up a Trello board is a simple task. As it is just creating a to-do list of already finalised tasks that are copied from the WBS. Thus taking less time.</p> <p>Realistic: However, the Trello board maybe take 30 minutes as there may be issues the person setting up runs into such as wrong competed hours, remaining hours and cards completed.</p> <p>Pessimistic: Worst case scenario the task will take 120 minutes. As the Trello board doesn't work due to more complex glitches, in addition to this the WBS may not be finalised properly and the sub-task may not be broken down properly due to this extra time needs to be taken to fully go through all the tasks and create sub-task. Also, the addition of the client wanting another feature or a feature subtracts hence the product backlog and WBS for the sprint may have to be edited. This led to the Trello board having to be edited as well.</p>

Set up burn down chart	<p>O = 60 minutes M = $4 \times 120 = 480$ minutes P = 180 minutes</p> <p>E = $60 + 480 + 180$ E = $720 / 6$ E = 120 minutes</p>	<p>Optimistic: The burndown chart may take an hour to set up as it is an easy task to have done as it does generate after all the tasks have been added to the Trello board. Editing the settings of how the chart takes in data may take a little longer but not longer than an hour to perfect.</p> <p>Realistic: The setting up of the burndown chart may take 60 minutes longer thus realistically 120 minutes are estimated, which may be induced by the difficulties with setting up such as the burndown chart giving wrong currently finished hours, and days worked and hours completed. Hence another hour may be needed taken to edit chart data, thus as stated before 120 minutes of working time are estimated to be needed.</p> <p>Pessimistic: However, there may arise an issue where the chart data can't be edited on the day hence the project manager has to go into inspect and edit the data on the webpage where the Trello board so the data comes up with the correct days completed, hours completed and cards completed.</p>
Set up Git hub repository	<p>O = 30 minutes M = $4 \times 60 = 240$ minutes P = 90 minutes</p> <p>E = $30 + 240 + 90$ E = $360 / 6$ E 60 minutes</p>	<p>Optimistic: Optimist it is estimated this task will take 30 minutes as it just takes time to create the repository and set up the correct folders and add all the developers on the page. As well as those developers creating the repository on their desktop.</p> <p>Realistic: Realistically, however, the developers won't be as efficient in setting up the repository as first of all not all developers may be knowledgeable of GitHub, as well not have an account so time has to be taken to complete it to have those ready as well for those inexperienced developers to set up the site.</p> <p>Pessimistic: However pessimistically there may be errors in the set-up of the destroy causing files that are being added/pushed may not update with changes, or files may accidentally be deleted. As well certain developers may not have access to the respiratory, due to incorrect setup. Thus these errors take another 30 minutes to fix.</p>
Set up a schedule of the task to complete	<p>O = 30 minutes M = $4 \times 90 = 360$ minutes P = 120 minutes</p> <p>E = $30 + 360 + 120$</p>	<p>Optimistic: The development team can easily agree on what days what takes can start and then the day they are completed. This is understanding that all tasks have been decided and all the developers are free and also available in this process. Thus</p>

	$E = 510 / 6$ $E = 85 \text{ minutes}$	<p>30 minutes is estimated to be needed.</p> <p>Realistic: Realistically there may be an issue with developers not being fully available through the sprint hence there has to be time taken to alter this error. As well there may be contradictions ideas and opinions which the task can be completed. This hence leads to the time taken to discuss the schedule. Hence this may take 90 minutes.</p> <p>Pessimistic: Worst circumstance there maybe is an issue of the scrum team not being fully present in a stand-up meeting where the scheduling is being decided hence fewer people discuss and come to conclusion mutual. Leading to more time taken to complete the task. As well due to team members having other reasonability, they may not be 100% free which leads to the team having to go workaround and come up with different scheduling times. Which may increase the time taken to complete scheduling. Therefore takes 120 minutes to complete the task.</p>
Set up task allocation and team roles	$O = 30 \text{ minutes}$ $M = 4 \times 60 = 240 \text{ minutes}$ $P = 120 \text{ minutes}$ $E = 30 + 240 + 120$ $E = 390 / 6$ $E = 65 \text{ minutes}$	<p>Optimistic: In the case the scrum team can easily agree to what tasks are done and completed who. As quick as who wants to do what.</p> <p>Realistic: It is there may be an issue with team members not being fully available through the sprint hence there has to be time taken to alter this error and having to correctly allocate tasks. As well there may be a contradiction between who wants the task, hence there may be having to discuss carefully who is going to take the task.</p> <p>Pessimistic: Worst case there maybe be an issue of team members not being fully present in a stand-up meeting when the allocation of tasks is being done from the Trello board. Hence for absent team members either the team has to allocate on known skills or postpone the allocation. This leads to longer time spent on the allocation of the task than completion. In addition to this, there may be an issue that team members start to argue over the allocation of tasks. Hence the time has to be taken to stop meeting cool down and then start again in a few minutes. Therefore considering all these factors it may take 120 minutes in the worst case.</p>
Create software design report	$O = 60 \text{ minutes}$ $M = 4 \times 120 = 480 \text{ minutes}$ $P = 180 \text{ minutes}$	<p>Optimistic: If all developers are explained in UML, block diagram and sequence diagram this may take 60 minutes to complete if</p>

	$E = 60 + 480 + 180$ $E = 720 / 6$ $E = 120$ minutes	<p>developers concentrate and efficiently complete the task.</p> <p>Realistic: However, this task may take a little longer as maybe a developer is not able to efficiently complete the task as the task might be more difficult than thought.</p> <p>Pessimistic: Worst case the developer is not very knowledgeable or researched of selected software design diagram, hence time has to be taken to learn or revise how to complete a diagram and there may be possible mistakes in the diagram which need to be fixed.</p>
Total	480 minutes	

Design

Sub-task	Estimation of effort	Explanation
Design interface for Main dashboard	$O = 30$ minutes $M = 4 \times 60 = 240$ minutes $P = 120$ minutes $E = 30 + 240 + 120$ $E = 390 / 6$ $E = 65$ minutes	<p>Optimistic: Designing the main interface if easy decided may only take 30 minutes to develop as all the decided functionality is already has been allocated and finalised. Therefore, it's just implementing those features in an interface in a styled manner which counts that the designer is a fully qualified and efficient worker this will take 30 minutes.</p> <p>Realistic: However, 30 minutes maybe be so short of a period to complete the design of the interface as the designer has to check with multiple people for feedback as the progress of the design goes on. Such as the client and designers of the software are self for either clarification on a specific feature and its capabilities and is that reflected in the design. Also the client themselves to check if they are happy with the progress of the design. Also, the factor of the designer may be slower than intended hence needing another 30 minutes to complete the task.</p> <p>Pessimistic: If there are a lot of changes to the decided feature this may lead to multiple edits done to the design hence increasing the time spent on the design of the interface. As well the designer could be very slow at creating the design and indecisive in the selection of the look of the interface. Also, this may be due to the client not being happy with the first</p>

		<p>interaction of the design hence there need to be multiple copies of the design as well. In addition to this, there may be a communication breakdown between the designer and developers leading to errors/ mistakes in the design of the interface.</p>
<p>Design interface for To - do list section</p>	<p>O = 30 minutes M = 4 x 35 = 140 minutes P = 60 minutes</p> <p>E = 30 + 140 + 60 E = 230 / 6 E = 38.33 minutes → 40 minutes</p>	<p>Optimistic: Designing the to-do list interface if easy due to its smaller size and space taken up in the entire dashboard. Thus only takes 30 minutes to design all the decided functionality. Therefore, it just shows those features in an interface in a styled manner. As well if the designer is a fully qualified and efficient worker this will take 30 minutes.</p> <p>Realistic: Honestly however the optimistic estimation may be enough time to design the to-do interface. However, 5 minutes is given as a buffer to enable a little bit of time for feedback and to be very with the client and project manager about any changes that need to be made. Hence a total estimated time of 35 minutes.</p> <p>Pessimistic: It is predicted that if there are a lot of changes to the decided feature this may lead to multiple edits done to the design hence increasing the time spent on the design of the interface. As the client may be indecisive about how to want to have multiple design iterations. Also, there is a possibility for the design software to have issues and not work, leading to more time to cover these issues before actually being able to complete the design.</p>
<p>Design interface for the Trash widget/button</p>	<p>O = 10 minutes M = 4 x 30 = 120 minutes P = 60 minutes</p> <p>E = 10 + 120 + 60 E = 190 / 6 E = 31.67 minutes → 40 minutes</p>	<p>Optimistic: Designing the trash interface if easy decided may only take 10 minutes to develop as all the decided functionality is already has been allocated and finalised. In addition, the widget would take a smaller amount of time to draw up some drafts and then finalise those drafts. Hence not taking as much time to design.</p> <p>Realistic: If allowing the designer the time to work realistically designing the Trash interface is easy to decide may only take 30 minutes as stated before it's a small feature but the designer may take their time or have asked for feedback hence the time needed to verify the feature.</p> <p>Pessimistic: A worst-case is that the design of the trash button has to go through multiple interactions. In addition, the designer may be tired and hence less effective with work on the trash</p>

		button. With the possibility of the faulty design software which needs to be fixed before continuing, therefore, this task takes 60 minutes, in a worst-case situation.
Design interface for the daily calendar section	<p>O = 30 minutes M = 4 x 60 = 240 minutes P = 90 minutes</p> <p>E = 30 + 240 + 90 E = 360 / 6 E = 60 minutes</p>	<p>Optimistic: Designing of the interface daily calendar if easy decided may only take 30 minutes to develop as all the decided functionality is already has been allocated and finalised. In addition, as well the designer may take inspiration from the already existing calendar design.</p> <p>Realistic: However, this interface is medium in size compared to other interfaces hence this may take longer to draw up and finalise. Also, the extra time may allow the designer the time to work to perfect the design of the daily calendar interface hence an estimated time of 60 minutes. Moreover, the designer may take the time to discuss it between other designers and the client.</p> <p>Pessimistic: The worst case is that the design of the daily calendar has to go through multiple interactions. In addition, due to this, the designer becomes unmotivated, hence less effective with working on the daily calendar. With the possibility of the faulty design software which needs to be fixed before continuing, therefore, this task takes 90 minutes.</p>
Design interface for weekly calendar section	<p>O = 30 minutes M = 4 x 60 = 240 minutes P = 90 minutes</p> <p>E = 30 + 240 + 90 E = 360 / 6 E = 60 minutes</p>	<p>Optimistic: Designing of the interface weekly calendar can be easy completed take in a estimated 30 minutes. As all the decided functionality is already has been allocated and finalised. In addition, as well the designer may take inspiration from the already existing calendar design.</p> <p>Realistic: However, it is noted that realistically more time may allow the designer the time to work at a more reasonable pace designing the weekly calendar interface. Thus the estimated time of 60 minutes as stated should be enough time to design and very before finalising the design.</p> <p>Pessimistic: The worst case is that the design of the weekly calendar has to go through multiple versions as the client might want to add more features or have a specific design in mind. In addition, the designer may be tired and hence less effective with work on the weekly calendar. With the possibility of the faulty design software which needs to be fixed before continuing, therefore, this task takes 90 minutes, in a worst-case situation.</p>

Design interface for Monthly calendar section	$O = 30 \text{ minutes}$ $M = 4 \times 60 = 240 \text{ minutes}$ $P = 90 \text{ minutes}$ $E = 30 + 240 + 90$ $E = 360 / 6$ $E = 60 \text{ minutes}$	<p>Optimistic: Designing of the interface monthly calendar if easy decided may only take 30 minutes to develop as all the decided functionality is already has been allocated and finalised. In addition, as well the designer may take inspiration from the already existing calendar design.</p> <p>Realistic: However, this interface is larger with more features to design, hence this may take longer to develop. Also, the extra time may allow the designer the time to work realistically designing the monthly calendar interface taking 60 minutes as stated before it's a larger interface feature but the designer may take their time or have asked for feedback to verify the design.</p> <p>Pessimistic: Worst case, the design of the monthly calendar has to go through multiple interactions. In addition, the designer may be tired and hence less effective with work on the monthly calendar. With the possibility of the faulty design software which needs to be fixed before continuing, therefore, this task takes 90 minutes, in a worst-case situation.</p>
Total	6760 minutes	

F1: Add task

Sub-task	Estimation of effort	Explanation
Create and implement Task class	$O = 90 \text{ minutes}$ $M = 4 \times 120 = 480 \text{ minutes}$ $P = 240 \text{ minutes}$ $E = 90 + 480 + 240$ $E = 810 / 6$ $E = 135 \text{ minutes}$	<p>Optimistic: It is estimated that the task may take 90 minutes as if the developer is experienced and efficiently works on class implementation.</p> <p>Realistic: However the task itself may take longer as the developer may not be able to work as quickly as through. One either due to issues caused by an error occurring at debugging which takes longer to fix those errors before moving on with coding. Also, there may be issues where the developer has to look up how to code a certain bit adding to the time needed.</p> <p>Pessimistic: There may be any issues with development as the class maybe have not been implemented properly lacking certain functionality and also the code may not be read so if the</p>

		<p>other member on the development team goes to add to the code there is a gap of understanding on the functionality of certain bits of the code. In addition to that the developers working on class implementation maybe not be as knowledgeable of OOP or used it in their coding beforehand. Thus there may be time taken to refresh or learn the class implementation. This leads to more time needed to complete tasks as well the arising of errors will affect the productivity of the code development.</p>
Create and implement productivity dashboard page GUI interface	<p>O = 60 minutes M = 4 x 120 = 480 minutes P = 180 minutes</p> <p>E = 60 + 480 + 180 E = 720 / 6 E = 120 minutes</p>	<p>Optimistic: A GUI interface is seemingly easy to implement with a strong and understandable interface design for the developers to program. Therefore 60 minutes in the best-case scenario would be enough proficient time to implement the dashboard page as a GUI interface.</p> <p>Realistic: However, in reality, there are many things that can occur that lead to the developer taking another 60 minutes to code the task. Thus 120 minutes in total, as the developer may be tired hence working slower. Also, the developer may need to refer to the GUI documentation to revisit how to implement something in the GUI.</p> <p>Pessimistic: The worst-case scenario would be that the GUI language used in the coding of the software is not used by the developers. This leads to the developers having to learn or having to decide on a different GUI. Which may take another hour, in addition to the real working time needed to be taken to implement the interface. Therefore leads to 180 of estimated working time in the worst-case scenario.</p>
Create a To-do list GUI widget interface	<p>O = 60 minutes M = 4 x 90 = 360 minutes P = 120 minutes</p> <p>E = 60 + 360 + 120 E = 540 / 6 E = 90 minutes</p>	<p>Optimistic: A GUI interface is seemingly easy to implement with a strong and understandable interface design for the developers to program. Therefore 60 minutes in the best-case scenario would be enough proficient time to implement the dashboard page as a GUI interface.</p> <p>Realistic: However reality there are many things that can occur that lead to the developer taking 90 minutes to code the task. As the developer may be tired hence working slower. Also, the developer may need to refer to the GUI documentation to revisit how to implement something in the GUI.</p> <p>Pessimistic: The worst-case scenario would be that the GUI implementation used in coding the software is not used by</p>

		<p>the developers. This leads to the developers having learned or having to decide on a different GUI. Which may take another hour, in addition to the real working time needed to be taken to implement the interface. Therefore leads to 120 of estimated working time in the worst-case scenario. Also, the designer and developer may have a disagreement on how the interface may look, leading to the time taken to complete the GUI taking longer due to this.</p>
<p>Create Add Task button GUI interface</p>	<p>O = 15 minutes M = 4 x 30 = 120 minutes P = 60 minutes</p> <p>E = 15 + 120 + 60 E = 195 / 6 E = 32.5 minutes</p>	<p>Optimistic: A GUI interface is seemingly a small and a portion of the code to implement. In addition, If the developer has a strong understanding of the design. Therefore 15 minutes in the best-case scenario would be enough proficient time to implement the Add-task button GUI interface.</p> <p>Realistic: However, in reality, there are many things that can occur that lead to the developer taking 30 minutes to code the task. As the developer may be tired hence working slower. Also, the developer may need to refer to the GUI documentation to revisit how to implement something in the GUI.</p> <p>Pessimistic: The worst-case scenario would be that the GUI implementation used in coding the software is not used by the developers. This leads to the developers having learned or having to decide on a different GUI. Which may take another hour, in addition to the real working time needed to be taken to implement the add task interface. Therefore leads to 60 estimated working time in the worst-case scenario. Also, the designer and developer may have a disagreement on how the interface may look, leading to the time taken to complete the GUI taking longer due to this.</p>
<p>Create and implement SQL query to link to the database</p>	<p>O = 30 minutes M = 4 x 35 = 140 minutes P = 90 minutes</p> <p>E = 30 + 140 + 90 E = 260 / 6 E = 43.33 minutes</p>	<p>Optimistic: The estimated time decided is 30 minutes. As the task itself shouldn't take long the code needed should be fairly small compared to other coding task such as class implementation.</p> <p>Realistic: There may be errors and the SQL file not work encountered during the development of the query. This leads to taking 5 minutes more to alter and fix errors. Which gives more leeway to those errors.</p> <p>Pessimistic: However it is estimated that the worst-case scenario would</p>

		be the task taking 90 minutes either due to a failing query due to errors hence taking a lot of time to understand and fix those errors, Additionally, the developers may be either could not be knowledgeable of SQL or need to relearn bits of it hence taking this time may lead to an increased time taken to complete the task.
Total	420.83 minutes	

F2: Remove task

Sub-task	Estimation of effort	Explanation
Create removed class	<p>O = 30 minutes M = 4 x 120 = 480 minutes P = 180 minutes</p> <p>E = 30 + 480 + 180 E = 690 / 6 E = 115 minutes</p>	<p>Optimistic: It is estimated that the task may take 30 minutes if the developer is experienced and efficiently works on class implementation. Also, the class has implemented a deconstruct and then implementing other functions to remove this deleted task.</p> <p>Realistic: However, the task itself may take longer as the developer may not be able to work as quickly as through. One due to issues caused by an error occurring at debugging which takes longer to fix those errors before moving on with coding. Also, there may be issues where the developer must look up how to code a certain bit adding to the time needed.</p> <p>Pessimistic: There may be any issues with development as the class maybe have not been implemented properly lacking certain functionality and also the code may not be read so if the other member on the development team goes to add to the code there is a gap of understanding on the functionality of certain bits of the code. In addition to that the developers working on class implementation maybe not be as knowledgeable of OOP or used it in their coding beforehand. Thus there may be time taken to refresh or learn the class implementation. This leads to more time needed to complete tasks as well the arising of errors will affect the productivity of the code development.</p>
Create delete page (Trash) with deleted task GUI interface widget	<p>O = 30 minutes M = 4 x 60 = 240 minutes P = 180 minutes</p> <p>E = 30 + 240 + 180 E = 450 / 6</p>	<p>Optimistic: A GUI interface is seemingly easy to implement with a strong and understandable interface design for the developers to program. Therefore 30 minutes in the best-case scenario would be enough proficient time to implement the deleted task GUI interface.</p>

	<p>E = 75 minutes</p>	<p>Realistic: However reality there are many things that can occur that lead to the developer taking another 60 minutes to code the task. As the developer may be tired hence working slower. Also, the developer may need to refer to the GUI documentation to revisit how to implement something in the GUI.</p> <p>Pessimistic: The worst-case scenario would be that the GUI implementation used in coding the software is not used by the developers. This leads to the developers having learned or having to decide on a different GUI. Which may take another hour, in addition to the real working time needed to be taken to implement the interface. Therefore, leads to 180 of estimated working time in the worst-case scenario.</p>
Create Trash GUI interface widget	<p>O = 30 minutes M = 4 x 40 = 160 minutes P = 120 minutes</p> <p>E = 30 + 160 + 120 E = 310 / 6 E = 51.67 minutes → 60 minutes</p>	<p>Optimistic: This widget in the best case will take 30 minutes to implement the GUI code as the trash icon itself is small and if designed to simply will only take a couple of lines.</p> <p>Realistic: However, realistically the time taken to complete this implementation of this widget may take another 10 minutes hence 40 minutes. This may be caused by human efficiency being slow to complete the task and also the error arising in the code which may need to be fixed.</p> <p>Pessimistic: In the worst case it is estimated that the time taken to complete the task will take 120 minutes of coding the GUI. This is maybe caused by the developer's code being very unorganised and incomplete hence the widget may not look as intended. Therefore, this has to be edited and fixed which will take extra time.</p>
Create SQL query to link to database	<p>O = 30 minutes M = 4 x 40 = 160 minutes P = 90 minutes</p> <p>E = 30 + 160 + 90 E = 280 / 6 E = 46.67 minutes</p>	<p>Optimistic: The estimated time decided is 30 minutes. As the task, itself shouldn't take long the code needed should be fairly small compared to another coding task such as class implementation.</p> <p>Realistic: There may be errors and the SQL file not work encountered during the development of the query. This leads to taking 5 minutes more to alter and fix errors. Which gives more leeway to those errors.</p>

		Pessimistic: However it is estimated that the worst-case scenario would be the task taking 90 minutes either due to a failing query due to errors hence taking a lot of time to understand and fix those errors, Additionally the developers maybe either could not be knowledgeable of SQL or need to relearn bits of it hence taking this time may lead to an increased time taken to complete the task.
Total	296.67minutes	

F3: Move task position in to-do list

Sub-task	Estimation of effort	Explanation
Create implement Position class	O = 60 minutes M = 4 x 120 = 480 minutes P = 180 minutes $E = 60 + 480 + 180$ $E = 720 / 6$ E = 120 minutes	Optimistic: It is estimated this task will take 60 minutes. Because the developer, in this case, may be experienced and efficiently works on class implementation but still takes an hour, due to its more complex feature of enabling a task to be identified as a block using a GUI code, which can be altered by moving the task using a grab function included in the class. Realistic: However, the task itself may, in reality, take longer as the function of grabbing an existing widget is more complex than just implementing an add or remove task class as they are just removing or adding data. Moreover, the developer may not be able to work as quickly as estimated in the best-case scenario. Either due to issues caused by an error occurring at debugging it takes longer to fix those errors before moving on with coding. Also, there may be issues where there is a gap in the knowledge of the developer. Pessimistic: Worst case scenario the task takes 180 minutes to program. As the developer is not qualified enough to complete the task and needs assistance to complete the programming required to implement the position class. Thus there may be time taken to refresh or learn the class implementation. However, even though developers may

		<p>be qualified and multiple developers are working on the class, random errors may occur due to incorrect code or the software used to code isn't working correctly. So further time has to be taken to rectify these issues before continuing.</p>
<p>Add grab widget GUI interface</p>	<p>O = 60 minutes M = 4 x 90 = 360 minutes P = 120 minutes</p> <p>E = 60 + 360 + 120 E = 540 / 6 E = 90 minutes</p>	<p>Optimistic:</p> <p>A GUI interface is seemingly easy to implement with a strong and understandable interface design provided by the designer for the developers to program as well as the relevant functions for the actual functionality has been implemented properly. Therefore 60 minutes in the best-case scenario would be enough proficient time to implement the grab widget GUI interface.</p> <p>Realistic:</p> <p>However in reality there are many things that can occur that lead to the developer taking another 30 minutes to code the task thus in total. As the developer may be tired hence working slower. Also, the developer may need to refer to the GUI documentation to revisit how to implement something in the grab interface as this feature is a more complex one than simply the look of the interface, but coding a moving task block.</p> <p>Pessimistic:</p> <p>The worst-case scenario would be that the GUI implementation used in coding the software is not used by the developers. This leads to the developers having to learn the selected GUI or having to decide on a different GUI. Which may take another 30, in addition to the real working time needed to be taken to implement the interface. Therefore leads to 120 minutes of estimated working time in the worst-case scenario.</p>
<p>Create SQL Query to link to database</p>	<p>O = 30 minutes M = 4 x 60 = 240 minutes P = 90 minutes</p> <p>E = 30 + 240 + 90 E = 360 / 6 E = 60 minutes</p>	<p>Optimistic:</p> <p>The estimated in the best case the time to complete the task will be 30 minutes as the task itself shouldn't take that long. As the code needed should be fairly small compared to another coding tasks such as class implementation. In addition to this if the tasks database would have already been implemented, now it's just creating a query that adds another set of data to the task database.</p>

		<p>Realistic:</p> <p>There may be errors and the SQL file not work encountered during the development of the query. Also, there may be a possibility that the database itself was not set up correctly leading to that needing to be edited before. Which lead to another 30 minutes spent on that issue. Therefore 60 minutes in total.</p> <p>Pessimistic:</p> <p>However, it is estimated in the worst-case scenario that the SQL query to link the database will take 90 minutes. Due to errors in SQL code, the database is not set up correctly or also the developer in charge of the back end is not being as knowledgeable in SQL hence time might be taken to re-learn. Also, there is the issue of sometimes SQL not functioning how the developers want it to. So the code might have to be re-done in that case.</p>
Total	270 minutes	

Testing

Sub-task	Estimation of effort	Explanation
Create and execute Test cases for F1	<p>O = 60 minutes M = 4 x 180 = 720 minutes P = 240 minutes</p> <p>E = 60 + 720 + 240 E = 1020 / 6 E = 170 minutes</p>	<p>Optimistic:</p> <p>Optimistically it is estimated that testing of the following backlog item would only take 60 minutes at maximum. As in the best-case scenario, all the tests pass and the code has no errors working as intended. The time taken is to only account for the time needed to implement the test cases.</p> <p>Realistic:</p> <p>Realistically though there are going to be failing test cases which may take another 60 minutes to understand its causes and another 60 minutes to fix those sources of error. Thus leading to another 120 minutes added to the optimistic estimated time. In addition, there is the issue of human error and human efficiency. The developer doing the test cases may be slow at the task or tired or not fully concentrating on the task leading to errors or longer work time, hence these factors must be considered during the realistic estimation.</p>

		<p>Pessimistic:</p> <p>The worst case situation is that the test cases all fail hence in total 240 minutes must be taken to understand and locate the error. Then to find the solution to those errors as well test the test cases again and see if it passes. This process may have to repeat many times until all the test cases pass. Also, the factor of human efficacy plays a part in how fast the testing of this backlog item occurs. As well it has to be taken into consideration that the backlog item code itself may have not been completed hence the time taken to fill gaps and then run the tests.</p>
Create and execute Test cases for F2	<p>O = 60 minutes M = 4 x 90 = 360 minutes P = 120 minutes</p> <p>E = 60 + 360 + 120 E = 540 / 6 E = 90 minutes</p>	<p>Optimistic: optimistically the predicted time to test of backlog item F2 would only take 60 minutes. As in this situation, all tests are passed and tasks are able to be deleted and stay deleted and additionally are removed from the task database as well. Hence the time taken is to only account for the time needed to implement the test cases.</p> <p>Realistic: However if all factors are considered it is estimated that the time taken to complete testing of F2, would be 90 minutes as realistically there will be errors that arise and need to be fixed during testing as these errors do take time to alter and test again. Moreover, the developer may need assistance in fixing certain errors as they may have tried to fix this problem couple of times but still haven't gotten to the solution. Hence multiple developers may need to look at the issue this increasing the time taken to fill them in and orientate them with the testing task. Therefore all these factors may lead to increased testing time.</p> <p>Pessimistic: A worst-case situation is that the remove task feature does not work at all. But there are no errors that are detected in the test cases. However, the feature on the application does not work as intended. Hence the testing process has to be re-done until the test highlight the issue with the error. This leads to a total time of 120 minutes.</p>
Create and execute Test cases for F3	<p>O = 60 minutes M = 4 x 90 = 360 minutes P = 180 minutes</p> <p>E = 60 + 360 + 180 E = 600 / 6</p>	<p>Optimistic: Firstly all implemented tests for the methods and functionality for this feature, should run smoothly. Overall test cases should return with a minimum of one or two errors that can be altered and tested again in a time span of 60 minutes. Therefore a generally successful</p>

	E = 100 minutes	<p>overall test cases for this backlog item.</p> <p>Realistic: However there is the case that overall test cases are not an easy fix or just take an extended amount of time to complete. Hence instead of 60 minutes it most realistically will take 90 minutes as the developers may need assistance from other developers to complete the test case as the error especially for this backlog item of enabling the user to move a task block in the interface is more complex thus leading to more complex errors. Therefore, filling the other developers assisting on this test case will take another 30 minutes hence a total of 90 minutes for this task to be complete.</p> <p>Pessimistic: In the worst possible situation the estimated time taken will be 180 minutes. As the errors occurring take a while to alter, the code needed for the backlog item is pretty complex, causing more complex not easy to alter errors. There also may be a communication issue between backlog item developers and case testing developers. Due to backlog items, developers may not comment on code properly or not explain the functionality of each class and function properly. Also not give a reference for the coding style they used. Hence case testing team might have trouble understanding the code. This is especially for this backlog item as the code would be more complex, thus more area for confusion.</p>
Generate test case report	<p>O = 60 minutes M = 4 x 120 = 480 minutes P = 180 minutes</p> <p>E = 60 + 480 + 180 E = 720 / 6 E = 120 minutes</p>	<p>Optimistic: If all test cases are complete and documented including all issue and solutions for all test cases. This test case report can be completed in an hour's time or 60 minutes, with the tester efficiently writing the report. Therefore this ensures quality can be monitored for the project.</p> <p>Realistic: However, it needs to be taken into account that a report will go through a drafting process. Where the project manager has to look over the report to see if the report is clear, high quality and informative. This may lead to a longer estimated working time of 120 minutes due to this process before finally finalising the report. But this is still considering that all team members working on the report work efficiently.</p> <p>Pessimistic: In the worst-case situation, the time estimated is 180 minutes of working time needed. Firstly the project manager is not happy with the report so asks for</p>

		improvements before finalising the report. In addition to this, the test cases have the possibility of not being properly documented and the code for the test cases is not commented on well, so there would be difficult in writing a report. Moreover, further clarification is needed from the test case developers to be able to write a report, thus needing more time to write it.
Total	480 minutes	

Sprint 1 wrap up

Sub-task	Estimation of effort	Explanation
Present product to client	$O = 90$ minutes $M = 4 \times 120 = 480$ minutes $P = 180$ minutes $E = 90 + 480 + 180$ $E = 750 / 6$ $E = 125$ minutes	<p>Optimistic: In the best-case situation the client is happy with the product. Additionally, giving positive feedback isn't very elaborated and thus takes a shorter time to note down. The client also is able to understand and work on all software features after 30 minutes of demonstration and 30 minutes of them demoing it themselves. Thus 90 minutes is estimated to complete this task by showing each backlog item, which also would include another 30 minutes of letting the client discuss any additional features they may want to add or any questions they have with the development process.</p> <p>Realistic: But in reality, during the presentation of the product review, the client most likely will have more critical feedback, with the client asking questions about features in the software as well as the development process. Hence time will be needed to explain the answers to those questions. To buffer and expect the time needed, the estimated time is extended to 120 minutes. As time will be needed to receive feedback and possibility discuss a solution with the client and answer any questions they want. Additionally, the client may want to add or remove features that would need to be discussed further.</p> <p>Pessimistic: The worst-case scenario is that it takes 180 minutes to conduct the product presentation. As a situation can arise where there is a disagreement between the client and the scrum team in how the implementation of backlog items is completed and to what degree. As the client might want to have a feature added that is outside the scope of the product implementation, or may be difficult to implement in the time given to complete the sprint. Thus, this discussion will take time to come to a common solution. This problem is in addition to having critical feedback from clients, as well as them asking clarifying questions about backlog items.</p>

Generate project review report	<p>O = 60 minutes M = 4 x 120 = 480 minutes P = 180 minutes</p> <p>E = 60 + 480 + 180 E = 720 / 6 E = 120 minutes</p>	<p>Optimistic: In the best-case scenario all backlog items are completed. Thus no further documentation citing why certain or all backlog items were not completed needs to be done. Hence management team just has to outline the feedback comments the client made for each backlog item in the sprint, as well as document the meeting minutes of the product review. Therefore, completing this product review report in 60 minutes.</p> <p>Realistic: Realistically however it has to be considered that during a sprint not everything goes to plan as not all backlog items are going to be completed to the highest quality, as the client will maybe have some things they want to change or improve about the product. Hence writing those comments and reporting on these comments may take more time than a simple sentence of positive feedback, which can be easily analysed and explained in the report. Furthermore, the product review report will go through iterations until the project manager will mark it as complete on the Trello board. Therefore in considering all these factors the time estimated will be 120 minutes.</p> <p>Pessimistic: The worst scenario is the product review takes 180 minutes to complete. Due to extra documentation to justify why certain backlog items were not complete and issue the management team faced during development. As well as comments from clients of feedback on the backlog items.</p>
Generate Process review report	<p>O = 60 minutes M = 4 x 120 = 480 minutes P = 180 minutes</p> <p>E = 60 + 480 + 180 E = 720 / 6 E = 120 minutes</p>	<p>Optimistic: The development team is able to come to a collective decision about the things that worked well and didn't work well in there process of completing the sprint. Which will have a time breakdown of 30 minutes used to discuss and another 30 minutes to document and categorise the things discussed into things that went well and things that didn't, and also outline possible solutions. I total the time taken to complete the process review report in the best-case scenario would be 60 minutes.</p> <p>Realistic: However in reality isn't as simple the write up this process review report as stated in the justification for optimistic estimation, as there are many factors to take into account when estimating the time needed. Such as there can be disagreement on what process during the sprint worked well and which didn't. Therefore extending the estimated time to 120 minutes. Realistically in addition to team disagreements, it has to be taken into account the issue of developers being burnt out as well, affecting the process review quality as it is completed at the end of the sprint. Hence as the management team may struggle to put out more quality work than normal, therefore time needs to be taken to</p>

		<p>alter and increase the quality of the process review report.</p> <p>Pessimistic: Pessimistically estimated time taken on task would be 180 minutes. This can be due to an incomplete backlog of items leading to an incomplete sprint. Thus the report must include the issues and solutions which help improve how the next sprint is completed. Also planning how to complete the remaining backlog items that have been promised to the client. This firstly would take more time to discuss between the management team and the client about how they want to go about completing the unfinished backlog items, as well as documentation of all this information in the respective heading of things that worked and didn't work.</p>
Total	365 minutes	