Cost-Importance Table

Analyzing all the data collected in the Cognitive Walkthrough Sheets

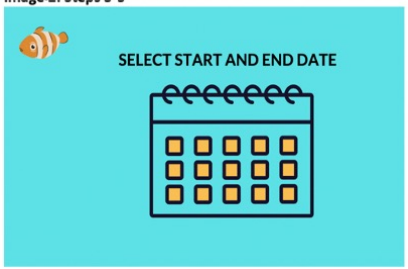
Analyzing your coding sheets from the Cognitive Walkthrough

- Initial steps:
- Create a table: a cost-importance table (you'll do an adapted version of this for your final analysis)
  - Regardless of how you get your list of UX problems
    - Rigorous lab-based testing or rapid evaluation techniques
  - Usually, cannot afford to fix them all
  - Need to prioritize
  - Use a cost-importance table

Reminder of your Cognitive Walkthrough Sheets

- How to take all the data from these sheets and translate into usable information to help inform which changes you need to make.

Prototype Page 2

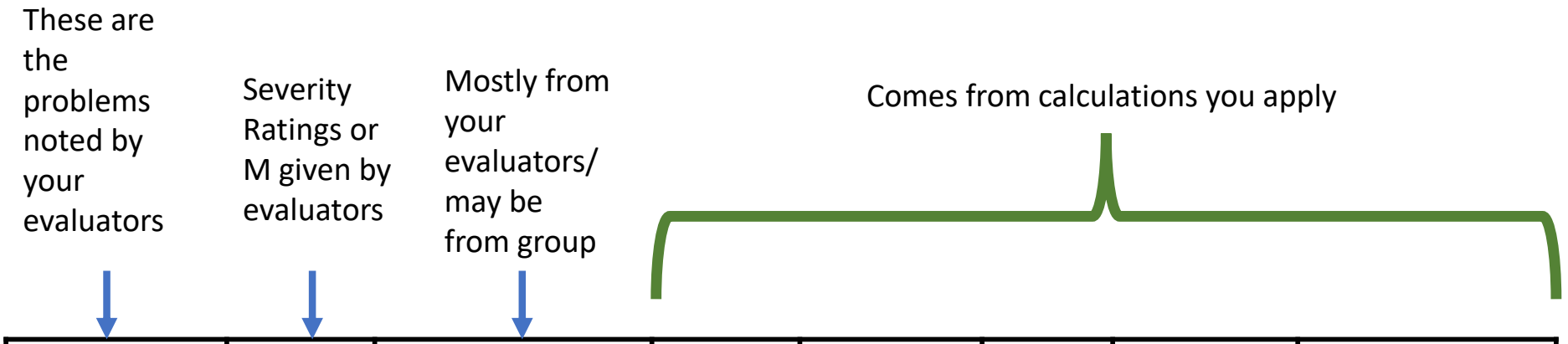


Step 2: Abby fills in the start and end date to set up the dates for the feeding schedule. [UC steps 3-5]

Question	Yes (reason/s)	No (reason/s)	Maybe (reason/s)	Severity Rating (1-5)
Will the correct action be sufficiently evident to Abby? ("Know what to do?" -Will the Abby know what to do to achieve the task?)			Abby might not know that the calendar is to add days to feed fish. Could add a title beyond the simple instructions.	1
Will the Abby notice that the correct action is available? ("See how to do it" - Can users see the button or menu item that they should use for the next action? Is it apparent when needed?)			Abby might not know how to add dates. Could add better instructions under calendar of have a ? Icon for help.	~~3~~ M
Will the Abby associate and interpret the response from the action correctly ("Understand correct action/not correction" - will users know from the feedback that they have made a correct or incorrect choice of action?)		The next screen is very basic. The user may not know if they properly selected the correct dates. Perhaps join the 2 screens into one, ask frequency while keeping range of dates highlighted on calendar		~~4~~ M

Cost-importance table

Cost-importance table – several different columns



Problem	Imp.	Solution/s	Cost	Priority ratio	Prio. rank	Cuml. cost	Resolution

Problem Description
Example from Textbook:

- User had decided on an event to buy tickets for and had made choices for event, date, venue, seats, price
  - But did not realize that it was necessary to click on “Submit” button to seal the deal and move to screen for making payment
  - This is what we call the “do it” action (they didn't know how to do it)

Problem description

- User did not realize that it was necessary to click on “Submit” button to move to screen for making payment for their event selection.

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label “Submit” to proceed to payment part of the purchasing transaction							

Step 2. Calculate the Importance to Fix

- This estimates how important it is to fix problem
  - This is independent of cost (i.e., don't consider cost at this point – just how important you think fixing the problem is).
  - In our table, this relates directly to the severity of a problem.

Importance to fix

- Estimate of importance to fix problem
  - Independent of cost
- Includes
  - Severity or criticality
  - Intangibles from management, marketing
- Importance = M: Must fix, regardless of cost
- Importance = 5: Most important problems to fix after "Must fix" category (Rate: 1 to 5) 5 is most critical

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution

Importance to fix = 5

- Feature is **mission critical. For example:**
  - Problem has major impact on user experience (e.g., the task is blocked)
  - Expected to occur frequently
  - Might cause costly errors

Importance to fix = 4

- In between a 3 and 5
  - E.g., It may be a medium impact problem will occur a lot for most users so nudge up to a 4

Importance to fix = 3-4

- Moderate impact problems
  - User can complete task, but with some difficulty (e.g., confusion and extra effort)

Importance to fix = 1-2

- Low impact problems
- Little impact on task performance or satisfaction (e.g., cosmetic)
  - But still worth listing

Importance adjustment factors

- Importance rating adjustment factors (up or down)
  - Occurs frequently, increase value slightly
  - Occurs rarely, decrease value slightly

Example: Importance to fix

- Our example problem (user didn't know you have to click on "Submit" to proceed)
  - Basic part of customer workflow
  - But not show-stopper
  - Initially Rate as 3

  - In our example problem, will be encountered by almost every user in almost every transaction
  - Promote rating.
  - Rate as 4

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction	4						

Step 3. Find a Solution

- Using the table, list the solution. If no solution made, you could come up with one as a group.
- You may have a couple of solutions or solutions that have a couple of points.

Example from Textbook: Solution

- For our example, obvious and inexpensive solution
  - Change label wording to better represent where interaction will go when user clicks on that button
  - Maybe “Proceed to payment” would make more sense to most users

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label “Submit” to proceed to payment part of the purchasing transaction	4	Change the label wording to “Proceed to Payment”					

Step 4. Figure out Cost to fix

- Estimated in person-hours
- Include cost of redesign
  - Design thinking and discussion and, sometimes, even some experimentation
  - Get help from software developers to estimate implementation costs (if appropriate)

Example: Cost to fix

- For our example (it could be hours to fix)
  - Cost = <1 hour so add 1
  - If it would take 3 hours to fix put down a 3

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction	4	Change the label wording to "Proceed to Payment"	1				

Step 5. Priority Ratio

- Want to
  - Reward high importance
  - Penalize high cost
- **Simple ratio of importance to cost**
  - High importance will boost up priority
  - High cost will bring it down
- Some factor to keep figure as integer (e.g., 1000)
- Priority ratio= (importance/cost) * 1000
- In our example: (4/1*1000)=4000

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction	4	Change the label wording to "Proceed to Payment"	1	4000			

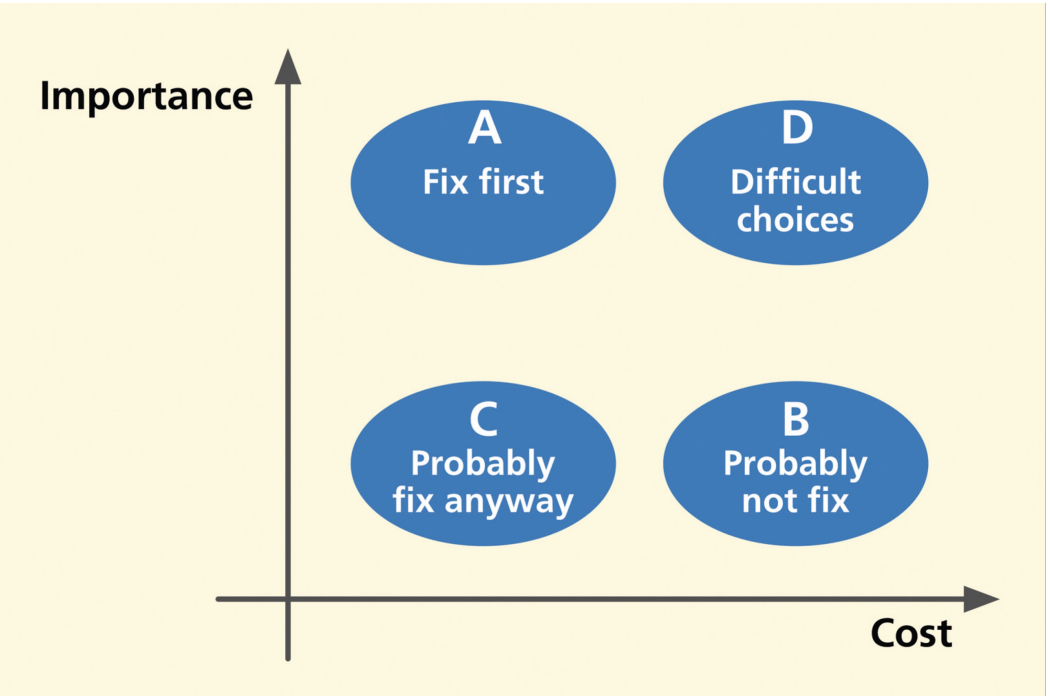
(4/1*1000)=4000

Note: if you have a M → then cost priority ratio is just M

Problem	Imp.	Solutions	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
User confused by the button label "Submit" to conclude ticket purchasing transaction	4	Change the label wording to "Proceed to Payment"	1	4000	(4/1*1000)=4000		
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed.	M	Move quantity information and label it	2	M			
Unsure of current date and what date he was purchasing tickets for	5	Add current date field and label all dates precisely	2	2500	(5/2*1000)=2500		
Users were concerned about their work being left for others to see	5	Add a timeout feature that clears the screens	3	1667			
User confused about "Theatre" on the "Choose a domain" screen. Thought it meant choosing a physical theater (as a venue), rather than the category of theatre arts.	3	Improve the wording to "Theatre Arts"	1	3000			
Ability to find events hampered by lack of a search capability	4	Design and implement a search function	40	100	(4/40*1000)=100		
Didn't recognize what geographical area theater information was being displayed for	4	Redesign graphical representation to show search radius	12	333			
Didn't like having a "Back" button on second screen since first screen was only a "Welcome"	2	Remove it	1	2000			
Transaction flow for purchasing tickets (group problem; see Table 16-8)	3	Establish a comprehensive and more flexible model of transaction flow and add labeling to explain it.	5	600			

Step 6. Set Priority rankings

- **Sort cost-importance table by priority ratio**
- **High importance does not always mean high cost**



Step 7. Calculate Cumulative Cost

- To calculate the cumulative cost, you take the cost of fixing that problem plus cost of fixing all problems above it

Pr	Imp.	Solutions	Cost	Prio. ratio	Prio. rank	Cumi cost	Resoluti on
Didn't recognize the "cou umber of tickets. As a result, us bout how many ti	M	Move quantity information and label it	2	M	1	2	
User confused by the butt nclude ticket purch	4	Change the label wording to "Proceed to Payment"	1	4000	2	3	
User confused about " se a domain" screen. Though ysical theater (as a venue), rathe heatre arts.	3	Improve the wording to "Theatre Arts"	1	3000	3	4	
Unsure of current date and what date he was purchasing tickets for	5	Add current date field and label all dates precisely	2	2500	4	6	
Didn't like having a "Back" button on second screen since first screen was only a "Welcome"	2	Remove it	1	2000	5	7	
Users were concerned about their work being left for others to see	5	Add a timeout feature that clears the screens	3	1667	6	10	
Transaction flow for purchasing tickets (group problem; see Table 16-8)	3	Establish a comprehensive and more flexible model of transaction flow and add labeling to explain it.	5	600	7	15	
Didn't recognize what geographical area theater information was being displayed for	4	Redesign graphical representation to show search radius	12	333	8	27	
Ability to find events hampered by lack of a search capability	4	Design and implement a search function	40	100	9	67	

For the Priority Rank, M move to the top, followed by the next highest priority ratio

4 = 1 + 1 + 2

10 = 3 + 1 + 2 + 1 + 1 + 2

Step 8. Line of affordability

- Determine your “resource limit,” in person-hours
- Draw “line of affordability” just above line where cumulative cost value exceeds resource limit

Problem	Imp.	Solutions	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed.	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to conclude ticket purchasing transaction	4	Change the label wording to "Proceed to Payment"	1	4000	2	3	
User confused about "Theatre" on the "Choose a domain" screen. Thought it meant choosing a physical theater (as a venue), rather than the category of theatre arts.	3	Improve the wording to "Theatre Arts"	1	3000	3	4	
Unsure of current date and what date he was purchasing tickets for	5	Add current date field and label all dates	2	2500	4	6	
Didn't like having a "Back" button on second screen since first screen was only a "Welcome"	2	Remove it	1	2000	5	7	
Users were concerned about their work being left for others to see	5	Add a timeout feature that clears the screens	3	1667	6	10	
Transaction flow for purchasing tickets (group problem; see Table 16-8)	3	Establish comprehensive and more flexible model of transaction flow and add labeling to explain	5	600	7	15	
Line of affordability (16 person-hours – two work days)							
Didn't recognize what geographical area theater information was being displayed for	4	Redesign graphical representation to show search radius	12	333	8	27	
Ability to find events hampered by lack of a search capability	4	Design and implement a search function	40	100	9	67	

Step 9. Drawing conclusions

- Resolution for each problem
- First, deal with “Must fix” problems
  - Hopefully, all above line of affordability

Possible resolutions

- Fix now
- Fix, time permitting
- Remand to “wait-and-see list”
- Table until next version
- Postpone indefinitely
  - Probably never get to fix

Problem	Imp.	Solutions	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed.	M	Move quantity information and label it	2	M	1	2	Fix in this version
User confused by the button label "Submit" to conclude ticket purchasing transaction	4	Change the label wording to "Proceed to Payment"	1	4000	2	3	Fix in this version
User confused about "Theatre" on the "Choose a domain" screen. Thought it meant choosing a physical theater (as a venue), rather than the category of theatre arts.	3	Improve the wording to "Theatre Arts"	1	3000	3	4	Fix in this version
Unsure of current date and what date he was purchasing tickets for	5	Add current date field and label all dates precisely	2	2500	4	6	Fix in this version
Didn't like having a "Back" button on second screen since first screen was only a "Welcome"	2	Remove it	1	2000	5	7	Fix in this version
Users were concerned about their work being left for others to see	5	Add a timeout feature that clears the screens	3	1667	6	10	Fix in this version
Transaction flow for purchasing tickets (group problem; see Table 16-8)	3	Establish a comprehensive and more flexible model of transaction flow and add labeling to explain it.	5	600	7	15	Fix in this version
Line of affordability (16 person-hours – two work days)							
Didn't recognize what geographical area theater information was being displayed for	4	Redesign graphical representation to show search radius	12	333	8	27	Wait until next version
Ability to find events hampered by lack of a search capability	4	Design and implement a search function	40	100	9	67	Wait until next version, or after that

Cost Importance Table

Using it to analyze your Cognitive Walkthroughs

Analyzing your coding sheets from the Cognitive Walkthrough

- Initial steps:
- Create a table: a cost-importance table (you'll do an adapted version of this for your final analysis)
  - Regardless of how you get your list of UX problems
    - Rigorous lab-based testing or rapid evaluation techniques
  - Usually, cannot afford to fix them all
  - Need to prioritize
  - Use a cost-importance table

Designing the Tables from the CW

- You will be given a template
- Create a cost-importance table for each task/feature you prototyped (add the task name to the first page for each table

Cost-importance Table – Task 1 (Add task name)

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Step 1: List Problems

- Go through all the cognitive walkthrough sheets **for each task** and list the problems as noted by the evaluators for each task in their own table. Write a short summary of the problem (e.g., see row one).
- Also, include a reference to the task and which step/sin the task name/use-case name, and note how many evaluators noted this problem, e.g. if two noted it, you could add 2/4 (sometimes all evaluators will note it and sometimes only one or two for example).

Note the reference to task, use-case steps and #of evaluators who noted it

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Step 1. Problem Description

- Add a problem identified by an evaluator (the 'maybe' or 'no' columns of our CW table)
- Abby may not know what she is supposed to do with only the calendar on the page (Task1, Steps 3-4, 1/2)
- Abby won't know how to add the dates for the feeder (from Task1, Steps3-5, Noted 2/2)
- Abby won't know that she added the dates in correctly (Task1, Steps3-4, 2/2)

Problems with some solutions

Step 2: Abby fills in the start and end date to set up the dates for the feeding schedule. [UC steps 3-5]

Question	Yes (reason/s)	No (reason/s)	Maybe (reason/s)	Severity Rating (1-5)
Will the correct action be sufficiently evident to Abby? ("Know what to do?" -Will the Abby know what to do to achieve the task?)			Abby might not know that the calendar is to add days to feed fish. Could add a title beyond the simple instructions.	1
Will the Abby notice that the correct action is available? ("See how to do it" - Can users see the button or menu item that they should use for the next action? Is it apparent when needed?)			Abby might not know how to add dates. Could add better instructions under calendar of have a ? Icon for help.	~~3~~ M
Will the Abby associate and interpret the response from the action correctly ("Understand correct action/not correction" - will users know from the feedback that they have made a correct or incorrect choice of action?)		The next screen is very basic. The user may not know if they properly selected the correct dates. Perhaps join the 2 screens into one, ask frequency while keeping range of dates highlighted on calendar		~~4~~ M

Problem	Imp.	Solution/s
Abby might know to use calendar to enter dates (Task1, Steps 3-4, 1/2)		
Abby might not know how to enter dates (T1, Steps3-5, 2/2)		
Abby may not know selected right dates – next screen very basic (T1, S3-4, 2/2		

Step 2. Add Importance

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

For each problem statement, give an importance value.

- Your evaluators gave 2-3 top fixes must fix that had to be done which represent M (must fix regardless of any other measure). Note, since you had two groups of evaluators, ***you may actually have 4-6 must fixes per task if each group of evaluators noted different issues. Include all as "M" in this case.***

Importance values include:

- Importance = M: Must fix, regardless (provided by the evaluators)
- Importance = x: (from 1 to 5 based on severity). **If you have one or more evaluators who noted the problem but gave it a different rating, take the average.** You may round up the average if you feel as a group that the rating is more serious than the average. The ratings are listed below:
  - Rating: 5 - feature is mission critical, could have major impact on user experience (e.g., task blocked), is expected to occur frequently and might cause costly errors.
  - Rating: 4 - in between a 5 and 3 where the problem may not be as critical (i.e., may not stop a user from doing a task) but may cause a lot of user frustration and happen very frequently.
  - Rating: 3 - user can complete task, but with some difficulty (e.g., confusion and extra effort)
  - Rating: 2 – in between a 1 and 3, for example the problem may have little impact on performance but may slightly affect user satisfaction
  - Rating: 1 - low impact problem/s, has little impact on task performance or satisfaction (e.g., cosmetic) but is still worth listing

Step 2. Calculate the Importance to Fix

- This estimates how important it is to fix problem
  - This is independent of cost (i.e., don't consider cost at this point – just how important you think fixing the problem is).
  - In our table, this relates directly to the severity of a problem.

Problems with some suggestions

Step 2: Abby fills in the start and end date to set up the dates for the feeding schedule. [UC steps 3-5]

Question	Yes (reason/s)	No (reason/s)	Maybe (reason/s)	Severity Rating (1-5)
Will the correct action be sufficiently evident to Abby? ("Know what to do?" -Will the Abby know what to do to achieve the task?)			Abby might not know that the calendar is to add days to feed fish. Could add a title beyond the simple instructions.	1
Will the Abby notice that the correct action is available? ("See how to do it" - Can users see the button or menu item that they should use for the next action? Is it apparent when needed?)			Abby might not know how to add dates. Could add better instructions under calendar of have a ? Icon for help.	~~3~~ M
Will the Abby associate and interpret the response from the action correctly ("Understand correct action/not correction" - will users know from the feedback that they have made a correct or incorrect choice of action?)		The next screen is very basic. The user may not know if they properly selected the correct dates. Perhaps join the 2 screens into one, ask frequency while keeping range of dates highlighted on calendar		~~4~~ M

Problem	Imp.	Solution/s
Abby might know to use calendar to enter dates (Task1, Steps 3-4, 1/2)	1	
Abby might not know how to enter dates (T1, Steps3-5, 2/2)	M	
Abby may not know selected right dates – next screen very basic (T1, S3-4, 2/2	M	

Step 3. List Solutions

- Sometimes the solution is quite simple (e.g., change label wording to better represent an interaction or to tell the user what happens when they click a button/link)
- Sometimes the solution may be more in depth (e.g., if a mobile device, divide information on one screen to more screens because it is too cluttered making it hard for a user, or rethink a metaphor that you used for a button, or you may have to redesign the look and feel of the screen, etc.).
- Some solutions will come from your evaluators and others will come from your group based on a discussion.

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Step 3. Find a Solution

- Using the table, list the solution. If no solution made, you could come up with one as a group.
- You may have a couple of solutions or solutions that have a couple of points.

Problems with some suggestions

Step 2: Abby fills in the start and end date to set up the dates for the feeding schedule. [UC steps 3-5]

Question	Yes (reason/s)	No (reason/s)	Maybe (reason/s)	Severity Rating (1-5)
Will the correct action be sufficiently evident to Abby? ("Know what to do?" -Will the Abby know what to do to achieve the task?)			Abby might not know that the calendar is to add days to feed fish. Could add a title beyond the simple instructions.	1
Will the Abby notice that the correct action is available? ("See how to do it" - Can users see the button or menu item that they should use for the next action? Is it apparent when needed?)			Abby might not know how to add dates. Could add better instructions under calendar of have a ? Icon for help.	3 M
Will the Abby associate and interpret the response from the action correctly ("Understand correct action/not correction" - will users know from the feedback that they have made a correct or incorrect choice of action?)		The next screen is very basic. The user may not know if they properly selected the correct dates. Perhaps join the 2 screens into one, ask frequency while keeping range of dates highlighted on calendar		4 M

Problem	Imp.	Solution/s
Abby might know to use calendar to enter dates (Task1, Steps 3-4, 1/2)	1	Add a title to the calendar
Abby might not know how to enter dates (T1, Steps3-5, 2/2)	M	Better instructions – add a ? That could be click for more help
Abby may not know selected right dates – next screen very basic (T1, S3-4, 2/2	M	Merge pages – keep date range highlighted while asking frequency

Step 5. Add Cost

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

- Normally, if you were in a company, the cost would be an estimated in person-hours for redesign. **But for our table, as a team you should a 3-point rating of cost:**
  - Rating of 1 – minimum time to make change
    - for problems that you think would take 10 minutes or less to do *on your LFP prototype* e.g., change the label of a button or move a button
  - Rating of 2 – more time needed to redesign
    - for problems that will take more than 10 minutes but less than 20 minutes or so – e.g., the solution may affect other features on your LFP page that may also require some redesign
  - Rating of 3 – more time needed (20+ minutes) because you need to think of the overall design
    - e.g., more than one LFP screen, add more information, inform the user of a mistake in a meaningful way (you'll probably only have a few (or even none for some tasks) identified as a 3, since we used a LFP).

Step 6: Priority Ratio & Priority Ranking

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Priority Ratio

- You want to reward high importance and penalize high-cost items. So high importance items will boost up priority and high cost will bring it down. Use the following ratio:

  *Priority ratio= (importance/cost) * 1000 [except for M]*

Priority Ranking

- Order (sort) the items in the table with highest ratio number first (note, **Importance "M"** are always at the top of the table, e.g., it doesn't matter how costly it is, these are the Must Fix from the evaluators).

Step 7. Cumulative Costs

Cost of fixing that problem plus cost of fixing all problems above it

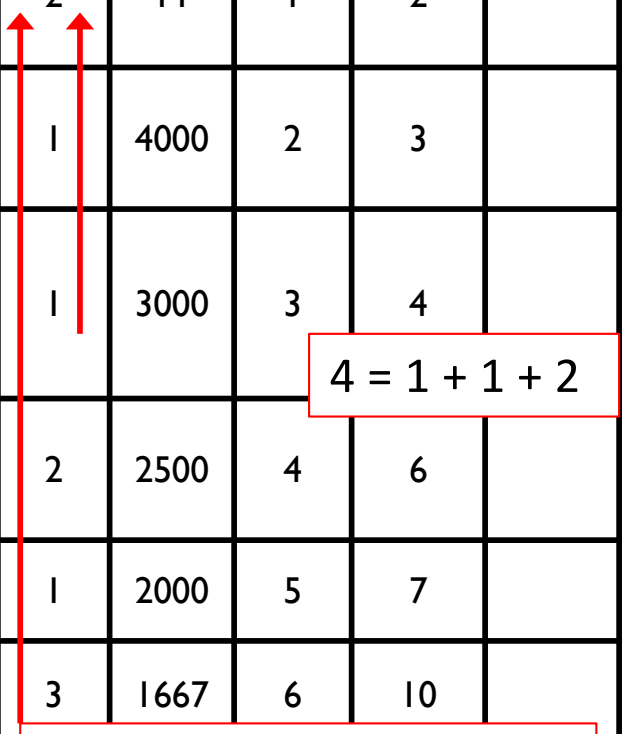
In this example:

- Row 1 has a cost of 2 (which is also the cumulative cost.
- Row 2 has a cost of 1, so its cumulative cost is 3 [2+1]

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Problem	Imp.	Solutions	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed.	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to conclude ticket purchasing transaction	4	Change the label wording to "Proceed to Payment"	1	4000	2	3	
User confused about "Theatre" on the "Choose a domain" screen. Thought it meant choosing a physical theater (as a venue), rather than the category of theatre arts.	3	Improve the wording to "Theatre Arts"	1	3000	3	4	
Unsure of current date and what date he was purchasing tickets for	5	Add current date field and label all dates precisely	2	2500	4	6	
Didn't like having a "Back" button on second screen since first screen was only a "Welcome"	2	Remove it	1	2000	5	7	
Users were concerned about their work being left for others to see	5	Add a timeout feature that clears the screens	3	1667	6	10	
Transaction flow for purchasing tickets (group problem; see Table 16-8)	3	Establish a comprehensive and more flexible model of transaction flow and add labeling to explain it.	5	600	7	15	
Didn't recognize what geographical area theater information was being displayed for	4	Redesign graphical representation to show search radius	12	333	8	27	
Ability to find events hampered by lack of a search capability	4	Design and implement a search function	40	100	9	67	



Step 8. Line of affordability

In a company this is often related to the total people hours given to make fixes.

Since you are doing tables for each task, you may not need a line of affordability for some of your tasks (i.e., 8 or less problems). If you have more than 10 problems, then you should set the line affordability between the **top 8-10 priorities based on feasibility.**

For A3, individual team members should adjust this to include 10-12 priorities (if there are 10+ problems) since each member will only focus on 3 tasks (their own choice if more than 3).

Add a row in your table after 8-10 problems indicating the line of affordability (note for your A4 this line will most likely be after 10-12 points)

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the "counter" as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	
User confused by the button label "Submit" to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to "Proceed to Payment"	1	4000	2	3 (2 + 1)	

Line of Affordability [e.g., top 10 changes]

Step 9. Resolutions

- In a company you would probably have the following resolutions: Fix now (above line of affordability), Fix - time permitting (just below the line of affordability), add to “wait-and-see list” (close to the line of affordability but low importance), table until next version, postpone indefinitely, probably never get to fix.
- For your table, you could have three resolutions: Fix now (above the line), fix-time permitting (just below the line), fix in future versions (at end of table). Again, for A4 depending on where the line affordability lies, if there are more than 10 problems then you will do some of the 'time-permitting' problems as well.

Cost-importance Table – Task 1 [Add task name]

Problem	Imp.	Solution(s)	Cost	Prio. ratio	Prio. rank	Cuml. cost	Resolution
Didn't recognize the “counter” as being for the number of tickets. As a result, user failed to even think about how many tickets he needed. **(Step/s X, from Task X, noted X times)**	M	Move quantity information and label it	2	M	1	2	Fix now
User confused by the button label “Submit” to proceed to payment part of the purchasing transaction **(Step/s X, from Task X, noted X times)**	4	Change the label wording to “Proceed to Payment”	1	4000	2	3 (2 + 1)	Fix now

For A4

- You will use the cost-importance table your group creates to help you design a higher-level prototype (HFP)
- A4 is individual – you will make your own design choices. Make sure you reference both the CI Table and design paradigms that we've looked at (e.g., Gestalts' Principles, CRAP, affordances). I wouldn't expect to see the same prototypes between team members.
- You will use your groups, tasks, scenarios and use-cases and recreate the prototypes at a higher level while making the changes outlined in the CI Table

- Cost-importance Table slides based on :
- Rex Hartson and Pardha Pyla. The UX Book: Agile UX Design for a Quality User Experience (2nd Ed) Morgan Kauffman, 2019. ISBN: 978-0-12-805342-3.