# Customer Risk

* Customer disobeys the application
  + You may not be able to safely load/unload your items or secure them properly inside the moving vehicle:
* Damage Items
  + You may not be able to pack your items safely enough
  + Traveling to destination, accident can occur
  + Your items may not have adequate insurance,
* Property Damage
  + damaged floors – scratched hardwood floors, torn carpets, broken tiles
  + damaged walls and/or door frames – chipped or dented walls, chipped or broken door frames, etc
  + damaged stairs – chipped staircase steps, broken banisters, etc.
* Personal Injuries
  + Persons with current injury (disenfranchises them)
  + Injured during the move
    - Strain your back, pull a muscle, sprain ankle
    - Trip or fall etc…
* Insufficient Experience
  + Unfamiliar with driving a bigger truck (potential accident due to lack of experience)
* Weak TOS (Get sued)
* Man Power
  + Might not have enough people to move the items in timely manner.
* Security Risk
  + Unattended item can get stolen.
  + Packed cars draw attention of thieves.
  + Losing items during the move.

# Technical Risk

* Artificial Intelligence Risk: Low data until self-improve.
* Server down time.
* Application performance
  + It’s important to ensure that any risk management plan encompasses user and partner expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the project to ensure that the work products are moving in the right direction.
* Application and system architecture
  + Taking a wrong turn on design can be a spell for disaster.
* New, unproven technology
* UI design being user friendly
* Inability to quickly implement user feedback due to technical complexity (-Jason)

<https://simplicable.com/new/technology-risk>

<https://moving.tips/diy-tips/major-risks-of-moving-by-yourself/>

https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects

<https://www.mymovingreviews.com/move/self-moving-risks/>

# Draft Take Two

## Technical Risk

* T-1: Artificial Intelligence prone to error due to low data.
* T-1 Mitigation: Preload test data via internet before application is released.
* T-2: UI Design not being user friendly. (move to customer risk, reword)
* T-2 Mitigation: Limit design changes per iteration and meet with stakeholders frequently.
* T-3: Inability to quicky implement user feedback due to technical complexity.
* T-3 Mitigation: Implement high level documentation and best coding practices to mitigation ease of future iterations.
* T-4: Current technology being implemented is new and unproven in the field.
* T-4 Mitigation: Conduct a beta phase to mitigate risk of critical errors upon release.

## Customer Risk

* C-1: End user disobeys the instructions given by the application.
* C-1 Mitigation: Implement an editing feature that allows the end user to change placement of items.
* C-2: User doesn’t understand how to use the application.
* C-2 Mitigation: Implement a tutorial on how to use the application and provide suggestions.
* ~~C-3: User is injured and files a lawsuit due to misuse of application.~~
* ~~C-3 Mitigation: Consult with lawyers to create a strong term of service to protect the end user and customer.~~
* C-4: User is disgruntled with the product.
* C-4 Mitigation: Implement a customer feedback feature that allows the users to disclose his/her issues with the applications.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Probability | | | | | | |
|  |  | Very Low | Low | Medium | High | Very High |
| Severity | Very High |  | C-3 |  |  | T-4 |
| High |  |  | T-2, C-4 | T-1 |  |
| Medium |  |  | T-3, C-1 |  |  |
| Low |  | C-2 |  |  |  |
| Very Low |  |  |  |  |  |
| Acceptable: Risk is acceptable level.  Permissible: Risk is okay for now and can be fixed at a later date.  Considerable: Risk is noted and will be fixed in the next iteration.  Catastrophic: Product is placed on hold until issue is fixed. | | | | | | |

This is subject to change… I kind of just threw these in there. Plus my chart looks like shi….

https://www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/risk-management/risk-mitigation-planning-implementation-and-progress-monitoring

<https://www.smashingmagazine.com/2020/05/reducing-design-risk/>

# Draft Take Three (Remember… be a word smith… become the word smith.. i… am… the word smith!)

## Technical Risk

* T-1: Artificial Intelligence prone to error due to low data.
* T-1 Mitigation: Preload test data in the beta phase with test users.
* T-2: Reduced functionality of application due to time constants and budget.
* T-2 Mitigation: Implement key features in the application for first iteration.
* ~~T-2: UI Design not being user friendly. (move to customer risk, reword)~~
* ~~T-2 Mitigation: Limit design changes per iteration and meet with stakeholders frequently.~~
* T-3: Inability to quicky implement user feedback due to technical complexity.
* T-3 Mitigation: Implement high level documentation and best coding practices to mitigation ease of future iterations.
* T-4: Current technology being implemented is new and unproven in the field.
* T-4 Mitigation: Conduct a beta phase to mitigate risk of critical errors upon release.
* T-5: Key developer leaves the project and takes critical information with him/her.
* T-5 Mitigation: Ensure team members collaborate and share knowledge equally/documented throughout the projects lift cycle.

## Customer Risk

* C-1: End user dissatisfied with the load plan generated by the application.
* C-1 Mitigation: Implement an editing feature that allows the end user to change placement of boxes.
* C-2: End user doesn’t understand how to use the application.
* C-2 Mitigation: Implement a tutorial on how to use the application and provide a help feature.
* C-3: End user finds UI challenging to operate.
* C-3 Mitigation: Meet with stakeholders to resolve issues to meet their needs.
* ~~C-3: User is injured and files a lawsuit due to misuse of application.~~
* ~~C-3 Mitigation: Consult with lawyers to create a strong term of service to protect the end user and customer.~~
* C-4: User is disgruntled with the product.
* C-4 Mitigation: Implement a customer feedback feature that allows the users to disclose his/her issues with the applications.
* C-5: User experience with certain parts of the application is low.
* C-5: Mitigation: Implement analytics for tracking when a user stop using the application.

# Draft Take Four

## Technical Risk

* T-1: Artificial Intelligence prone to error due to insufficient training data.
* T-1 Mitigation: Implementing a feedback loopback in the beta phase with test users.
* T-2: Reduced functionality of application due to time constants and budget.
* T-2 Mitigation: Implement key features in the application for first iteration.
* T-3: Inability to quicky implement user feedback due to technical complexity.
* T-3 Mitigation: Implement a feature for users to give feedback if the application operated correctly after completing a move.
* T-4: Current technology being implemented is new and unproven in the field.
* T-4 Mitigation: Conduct a beta phase to mitigate risk of critical errors upon release.
* T-5: Key developer leaves the project and takes critical information with him/her.
* T-5 Mitigation: Ensure team members collaborate and share knowledge equally throughout the projects lift cycle.

## Customer Risk

* C-1: End user is inexperienced with the application.
* C-1 Mitigation: Implement a tutorial on how to use the application and provide a help feature.
* C-2: End user finds UI challenging to operate.
* C-2: Mitigation: Implement analytics for tracking when a user stop using the application.
* C-3: End user is disgruntled with the product.
* C-3 Mitigation: Implement a customer feedback feature that allows the users to disclose his/her issues with the applications.