

## Clean Sweep Robotic Vacuum Cleaner & Sensor Simulator

Your company has assembled a small team of developers and business people to work on the implementation of the Clean Sweep robotic vacuum cleaner control system.

### 1 Control System

The control system integrates the Clean Sweep hardware components so that the vacuum can successfully navigate and clean a typical home.

#### 1.1 Navigation

The Clean Sweep cleans a typical home sweeping up any dirt and debris. Therefore it must be able to discover the basic floor plan of a home based on the input from its sensors. The Clean Sweep has four directional sensors around its perimeter that detect when it is about to encounter an obstacle. The control system must interpret the output from these sensors and stop the vacuum from running into any obstacles. Upon encountering an obstacle, the Clean Sweep must determine a new direction that prevents it from running into another obstacle. If the Clean Sweep is unable to move, it should shut itself down and wait for the customer to move it to a new location.

Under no circumstances should the Clean Sweep attempt to traverse stairs. A sensor on the bottom of the Clean Sweep can detect the edge of stairs or other declines. If this sensor detects anything, the control system should treat it as though it had encountered any other kind of obstacle.

The Clean Sweep can move in any direction so there is no need for the vacuum to change its orientation or turn around in order to navigate.

#### 1.2 Dirt Detection and Cleaning

As the Clean Sweep moves, a sensor determines if there is dirt to collect. If so, the control system should tell the vacuum to sweep up the dirt. The sensor is not able to tell how much dirt is present, only whether or not the current location is considered clean. Therefore multiple uses of the vacuum may be required to clean a particularly dirty area.

The Clean Sweep is equipped with a surface sensor that allows it to detect whether or not the surface it is currently traversing is bare floor, low-pile carpet, or high-pile carpet. It should automatically shift its cleaning mode between the surfaces accordingly.

Each use of the vacuum removes 1 unit of dirt. The Clean Sweep has a dirt-carrying capacity of 50 units. Once that capacity has been reached, it should return to its charging base and turn on its *Empty Me* indicator. The Clean Sweep doesn't stop cleaning until:

- It has visited every accessible location on the current floor.
- Its dirt capacity has been filled.
- It only has enough power remaining to return to its charging station. See *Power Management* for more details.

Each time the Clean Sweep begins a cleaning cycle, it automatically assumes that each location is dirty. A cleaning cycle is defined as the start of the next cleaning after all reachable areas have been cleaned. So, if the Clean Sweep is partially done cleaning a floor and has returned to its base station to recharge, it will assume that the current cleaning cycle is not yet complete.

### 1.3 Power Management

The Clean Sweep has a limited battery life of 250 units of charge. Each movement and vacuum operation requires units of charge depending on the surfaces being traversed:

- Bare floor - 1 unit
- Low-pile carpet - 2 units
- High-pile carpet - 3 units

The charge required for the Clean Sweep to move from location  $A$  to location  $B$  is the average of the required charge costs for the surfaces at the two locations. For example, if the Clean Sweep moves from a bare floor to low-pile carpet, then the charge required is 1.5 units. It costs the same amount of charge to clean the current location as it does to traverse that location.

The Clean Sweep should always return to its charging station before it runs out of power. Homeowners that come home and discover their robotic vacuum out of power in the middle of a room tend to be dissatisfied customers. As it returns to its charging station, the Clean Sweep will not perform any cleaning. Upon returning to its charging station, the Clean Sweep will automatically re-charge to full capacity. Once re-charged, it will resume cleaning until it detects that it has visited every accessible location on the current floor.

## 2 Registration

Registration includes the creation of an account and provides access to the Clean Sweep Portal.

Each Clean Sweep is registered with the company's website before it can be used. Once registered, the registrant has access to the Clean Sweep portal, which includes the ability to contact Customer Service for help with the unit as well as to visualize the vacuum's cleaning activities.

During registration, the user will need to create a password to associate with their account. The registrant's email address will be used to identify the account. The can also create their account through Google, Amazon, or Facebook.

Once the account has been created, the user can register their devices. Doing so requires that they either provide the serial number or scan the QR code printed on the same label as the serial number.

### 3 Clean Sweep Portal

The Clean Sweep Portal is the user's entryway to all things Clean Sweep.

#### 3.1 Scheduling

The Clean Sweep can be told to clean manually at the touch of a button on the unit itself. But by using the Clean Sweep Portal the user can define a schedule. In addition to starting the unit manually, the user can also define a one-time scheduled cleaning or a recurring schedule.

As part of each unit's configuration, the user will define the hours that the unit is permitted to clean. This allows the user to avoid having the unit cleaning at inconvenient times such as when people are home or sleeping.

##### 3.1.1 One-Button Cleaning

In this scenario the user requests that the unit begin cleaning and continue until either the cleaning is cancelled or until the floor is clean. If the request is made to start outside of the unit's preconfigured available hours, the user must confirm the action. If the user chooses **not** to begin cleaning immediately, then they should be guided to the *One-Time Cleaning* workflow.

The user may also determine when the unit should cease cleaning. The options are to continue cleaning until a) the floor is clean; or b) the cleaning time is outside of the unit's available cleaning hours.

For all intents and purposes this type of cleaning is the same as a One-Time Cleaning that is set to begin immediately.

##### 3.1.2 One-Time Cleaning

The user can also define a one-time schedule that tells the unit when to begin and end cleaning. This is basically the same as a One-Button Cleaning save that the cleaning cycle can be set to begin at some date in the future.

##### 3.1.3 Repeated Cleaning

In this workflow the user defines a repeated schedule. Conceptually this is the same as repeated One-Time Cleanings. In addition to the information gathered during the *One-Time Cleaning* the user can specify that the cleaning repeat on multiple days of the week. The user can also indicate that the cleaning should happen: a) a fixed number of times; b) through a particular date; or c) until the schedule is cancelled.

■ **Note:** Use your favorite event scheduling software as a model for these options.

#### 3.2 Visualization

After registration, the user can log in and visualize what the Clean Sweep has been doing. This includes the floor plan, the location of the recharging station, the path of the Clean Sweep as it vacuums, and information about the degree of dirt found at the various locations. This will provide a visual map of high-traffic areas, which Clean Sweep, Inc. may be able to use for future advertising initiatives.

## 4 Diagnostics & Troubleshooting

Since the Clean Sweep is a new product, it is important that technical support personnel are able to troubleshoot its operation. All of this happens within the Clean Sweep Portal.

### 4.0.1 Activity Log & Visualization

The Clean Sweep logs all of its raw activities as discrete events. These events relate to all major functions of the device and include the following data points:

- The current date and time
- Position of unit
- Direction of travel
- Surface detected
- Cleaning status
- Dirt detected
- Obstacles detected
- Current power
- Whether the unit is currently recharging

The logs are sent in real-time to the Clean Sweep Portal. While these logs can be reviewed by the registrant, they are primarily used by the Customer Support staff to troubleshoot problems.

## 5 Sensor Simulator

The Clean Sweep must maintain an internal map of the home. In order to write the control software, it is necessary to simulate the input of the Clean Sweep's sensor array as it might respond within actual homes. The Sensor Simulator provides this capability and acts as a virtual sensor array. These simulations will act as test data against which you can test your control software.

### 5.1 Floor Plan Representation

You will need to define a persistent floor plan layout. The layout file will need to represent different attributes of floor plan.

#### 5.1.1 Layout Files

To facilitate the testing of the Clean Sweep, the Sensor Simulator must be able to “play back” pre-defined floor plans. This means that the Sensor Simulator should be able to read a floor plan file and use that file to respond to requests from the Clean Sweep's control software.

The control software itself should have no knowledge of the floor plan layout. This file is only intended to allow the sensor simulator to emit controlled output based on interactions with the control software. The fact that the entire floor plan is available to the sensor simulator does not mean that the Clean Sweep has *a priori* knowledge of the home's layout. The Clean Sweep control system must discover the floor plan in real time just as the physical device would.

**Note:** It must be possible to change the layout file *in real-time* in order to simulate changing conditions in the house. For example, a door that was open might now be closed and vice versa or a new obstacle might now be present. Similarly the floor covering might change such as rugs being added or removed.

#### 5.1.2 Surfaces

The *surface sensor* detects the surface under the Clean Sweep. This determines how much power is required to move the vacuum. The surface sensor can identify the following floor coverings:

1. Bare floor (e.g. wood, linoleum, etc.)
2. Low-pile carpet
3. High-pile carpet

The *dirt sensor* can tell whether or not the current position must be cleaned. The simulator must be able to represent how many units of dirt are at a given location. While the sensor itself cannot tell exactly how many units are present, this will allow the simulator to continuously register a position as dirty until the appropriate number of vacuum operations have been performed.

#### 5.1.3 Navigation

The Clean Sweep has *navigation sensors* that allow it to move around the floor plan. The floor plan must identify where the Clean Sweep may navigate and when there are obstacles to its path. The floor plan must indicate whether there is a path from one point to another in the direction of travel or whether such navigation is blocked. Each path between two cells can contain the following:

1. Unknown. The Clean Sweep doesn't yet know what's in the specified direction.
2. Open. The sensor indicates that the path is open.
3. Obstacle. The sensor indicates that the path is obstructed.
4. Stairs. The sensor indicates that the path in that direction is blocked by stairs.

#### 5.1.4 Charging Stations

Each floor plan may contain one or more *charging stations*. These charging stations are used by the Clean Sweep to replenish its battery when it runs out of power. The Clean Sweep always knows about the charging station where it begins its cleaning process. It can discover others if it is within 2 cells of the charging station. The detection of the charging station happens whether or not the path to the station is actually clear. For example, it is possible that the Clean Sweep could detect a charging station in another room, but not be able to get to that station if the room's door is closed.

The control system must identify cells that contain charging stations. As the vacuum sweeps the floor, it will continually track which charging station is closest to its current position and move to that station when necessary.

Assignment Project Exam Help  
Add WeChat powcoder

<https://powcoder.com>

Add WeChat powcoder

## 5.2 Sample Floor Plan

This section provides a more complex floor plan. Figure 1 is an example of a basic floor plan. In this plan there are 3 bedrooms and a main hallway that will be cleaned.

Notice that the doors, represented by the small rectangles, can be either open or closed. Each time the Clean Sweep encounters that door, it will need to determine whether or not it is open. This means that it is technically possible for the customer to accidentally (or deliberately) shut the Clean Sweep in a room.

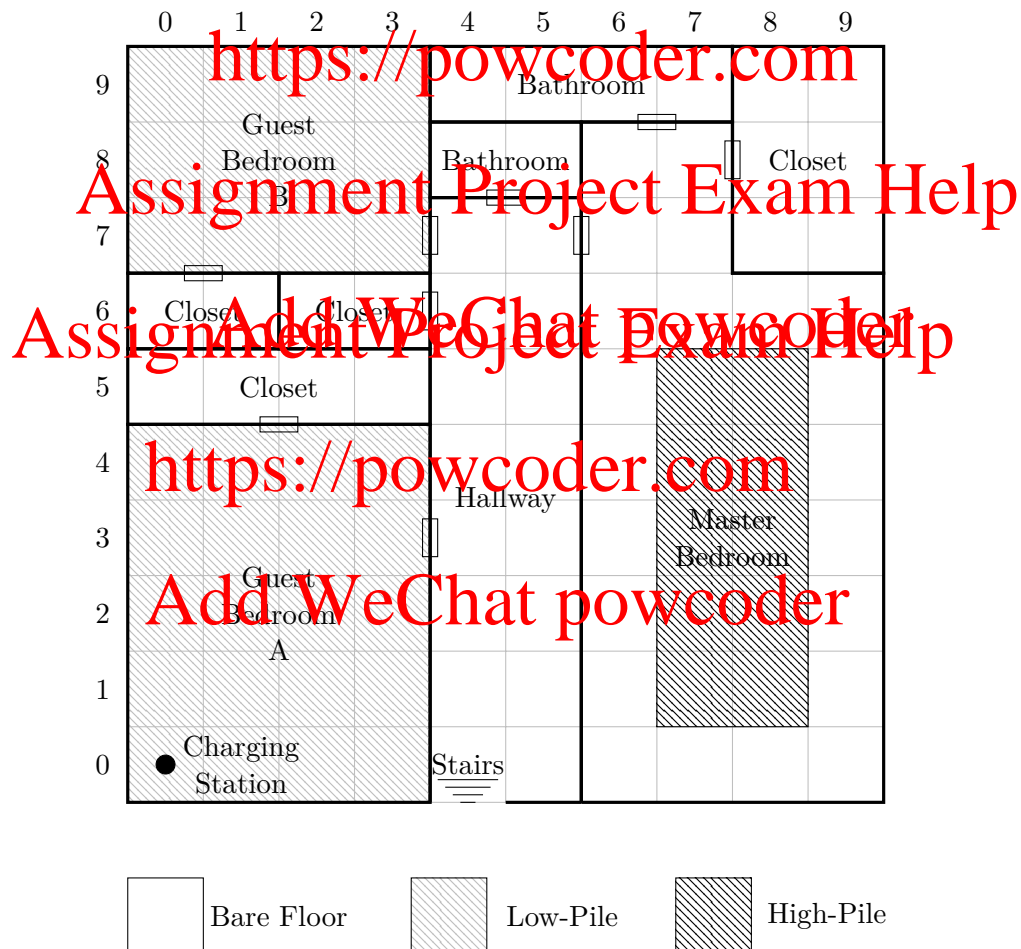


Figure 1: Sample Floor Plan

## 6 The Task

Your team will play the role of both “business” and “development”. The instructor will play the role of “management”. Your task is to construct and virtually the plan for the Clean Sweep’s development.

■ **Note:** *You will not actually be writing code in this class – it is not a programming class.*

### 6.1 Development

Your team is responsible for following Agile practices including working with the business to write user stories, estimating the development time, planning your releases, tracking your velocity and updating management’s expectations accordingly.

If there are any questions about project functionality, the instructor will provide the final determination.

■ **Note:** *It’s possible that the same question, asked by different teams, might get different answers. Instructors are devious that way.*

#### 6.1.1 Constraints

In order to keep everything in one place and to provide some consistency. We will adopt the same software across teams. While you may use your own software on the side, any submissions should be in the following packages:

- Trello – <https://trello.com/>



## 7 Activities and Ceremonies

You will use the basic activities and ceremonies of Scrum:

**Backlog Management.** You will need to keep your backlog groomed throughout the term. That means that you should always have a prioritized, estimated supply of stories. This will help sprint planning go much faster.

**Sprint Planning.** At the start of each sprint your team will carefully choose the stories that will add the greatest value for your minimum viable product (MVP). Plan on spending 1-2 hours per sprint for this activity.

**Estimation.** The team will need to agree on an estimation model for the various stories and tasks. Whatever model is selected, it must involve collaborative estimation. In other words, no one should be setting the estimate for a story or task without input from the other team members.

**Review.** At the end of each sprint, you will conduct a project review for the class that showcases what you've accomplished during the sprint. You would ordinarily plan on 30-60 minutes for this ceremony for a 2-week sprint, but since we're on a condensed schedule, plan for about 15-20 minutes instead.

**Retrospective.** At the end of each sprint, you will, as a team, conduct a sprint retrospective. You should plan on 30-60 minutes for this.

**Note:** The Daily Standup ceremony is absent. That's hard to maintain when we're all remote and on different schedules, so each team must decide what meeting cadence works best for them.

Add WeChat powcoder

## 8 Schedule

This project will run all term. You will deliver your project in phases using a simulation of the Scrum framework. The timeline and associated list of activities and deliverables is:

---

### Sprint 0 (weeks 2-4, 2019-SEP-21 - 2019-OCT-05)

**125 Points**

---

**Goal:** Backlog Creation

**Planning:** During sprint 0 you will construct the framework for your project including the epics, roles, and personas. You will use these during the subsequent sprints of the project.

1. Prepare your team templates as described in section 9.1 of this document.

2. Devise your approach to decomposing the Clean Sweep project. I've provided the initial themes for the project, but you may change these as you see fit.

3. Build out your product backlog in Trello. Focus on the epics and make sure that each epic is associated to one of the project themes from the template.

4. Define the roles and create personas that you wish to use to represent the different types of user who will work with the various Clean Sweep elements.

Add the personas to the "Roles and Personas" Trello list and create label for each of them.

**Review:** Your initial roles, personas, and epics in Trello.

**Retrospective:** Discuss how you can improve as a team. Identify what went well and what you as a team felt could be improved. Discuss any activities from your previous retrospective that the team wanted to change and discuss if the changes you made were beneficial.

**Evaluation:** This sprint will be evaluated on:

**50 pts. Backlog Management.** Your product backlog in Trello featuring epics.

**50 pts. Review.** The roles you identified, the epics you discovered, and a summary of your retrospective.

**25 pts. Retrospective.** The documentation and presentation of your team's lessons learned for the sprint and how you intend to apply those lessons to later sprints. Use the Trello template available here: [SE459 Sprint Retrospective Template](#).

**Submit:** A presentation that captures the major items described by the evaluation. Please include links to your Trello board. You only need to submit a single presentation for the team.

**Sprint 1 (weeks 4-6, 2019-OCT-05 - 2019-OCT-19)****125 Points**

---

**Goal:** Roadmap Planning

**Planning:** Author the initial set of stories for each epic, tying them to the personas you created in Sprint 0. Estimate each story using the “powers of two” scale and use these estimates to create a preliminary roadmap of 6 sprints. Complete the initial setup of the project plan within the Google sheet by following the steps in section 9.2 of this document.

**Review:** Your overall project plan, accounting, and breakdown.

**Retrospective:** Discuss how you can improve as a team. Identify what went well and what you as a team felt could be improved. Discuss any activities from your previous retrospective that the team wanted to change and discuss if the changes you made were beneficial.

**Evaluation:** This sprint will be evaluated on:

**50 pts. Backlog Management.** Your product backlog in Trello featuring your various sprints and the estimated stories assigned to them.

**50 pts. Review.** Your project plan, sprints, stories, and a summary of your retrospective.

**25 pts. Retrospective.** The documentation and presentation of your team’s lessons learned for the sprint and how you intend to apply those lessons to later sprints. Use the Trello template available here: [SE459 Sprint Retrospective Template](#).

**Submit:** A presentation that captures the major items described by the evaluation. Please include links to your Trello board. You only need to submit a single presentation for the team.

---

Sprint 2 (weeks 6-8, 2019-OCT-19 - 2019-NOV-02)125 Points

---

**Goal:** Scrum Simulation Game - Part 1

**Planning:** Complete the planning Google sheet and simulate development for sprints 1-3 of your plan by following the steps in section 9.3 of this document.

**Review:** The current state of your project including the accounting sheet and associated burndown charts showing your ideal, estimated, actual, and scope lines. Also show your updated sprint plan based your simulation.

**Retrospective:** Discuss how you can improve as a team. Identify what went well and what you as a team felt could be improved. Discuss any activities from your previous retrospective that the team wanted to change and discuss if the changes you made were beneficial.

**Evaluation:** This sprint will be evaluated on:

**50 pts. Backlog Management.** Your sprint backlog in Trello showing the current state of your project after the simulation. Your product backlog should also show new and removed stories.

**50 pts. Review.** Your updated project plan, accounting chart, burndowns, and a summary of your retrospective.

**25 pts. Retrospective.** The documentation and presentation of your team's lessons learned for the sprint and how you intend to apply those lessons to later sprints. Use the Trello template available here: [SE459 Sprint Retrospective Template](#).

**Submit:** A presentation that captures the major items described by the evaluation. Please include links to your Trello board. You only need to submit a single presentation for the team.

---

Sprint 3 (weeks 8-10, 2019-NOV-02 - 2019-NOV-16)125 Points

---

**Goal:** Scrum Simulation Game - Part 2

**Planning:** Complete the planning Google sheet and simulate development for sprints 4-6 of your plan by following the steps in section 9.3 of this document.

**Review:** The current state of your project including the accounting sheet and associated burndown charts showing your ideal, estimated, actual, and scope lines. Also show your updated sprint plan based your simulation.

**Retrospective:** Discuss how you can improve as a team. Identify what went well and what you as a team felt could be improved. Discuss any activities from your previous retrospective that the team wanted to change and discuss if the changes you made were beneficial.

**Evaluation:** This sprint will be evaluated on:

**50 pts. Backlog Management.** Your sprint backlog in Trello showing the current state of your project after the simulation. Your product backlog should also show new and removed stories.

**50 pts. Review.** Your updated project plan, accounting chart, burndowns, and a summary of your retrospective.

**25 pts. Retrospective.** The documentation and presentation of your team's lessons learned for the project. Use the Trello template available here: [SE459 Sprint Retrospective Template](#).

**Submit:** A presentation that captures the major items described by the evaluation. Please include links to your Trello board. You only need to submit a single presentation for the team.

## 9 Scrum Simulation Game

Since this is not a programming class, we're going to simulate the activities you would usually need to perform over the course of days or weeks. The goal is to acquaint you with some of the ceremonies and decisions that come from working within an agile framework without actually writing all of the code to make it happen.

### 9.1 The Templates

There are a few things you need in order to complete the project: A Trello board and the Google planning sheet.

#### 9.1.1 Trello Agile Project Board

For this project we're going to use Trello to help manage some of the day-to-day work of backlog grooming, sprint planning, etc. That means that each member of your team should create a Trello account if they don't already have one. Your team only needs one set of boards so you can choose which person will host them. That person can then invite the other team members to those boards.

Make a copy of the Trello board here: <https://powcoder.com>

You will update and customize this board through the remainder of the course.

#### 9.1.2 Google Planning Sheet

I have provided a Google sheet that each team will use a template. You only need one sheet for the team. The template is located here: <https://powcoder.com>

**Note:** Please make sure you make a copy of template and do not change the original. To make a copy, open the Google sheet and select "File > Make a copy".

### 9.2 The Setup

The initial setup of your planning sheet requires that your team complete the following steps:

1. Change your team name. Be as creative (or not) as you wish. Keep it clean.
2. Fill in the names of your team members in the "TEAM > BASE PRODUCTIVITY" section. If you have fewer than 6 members on your team, simply leave the other names as the defaults.

**Note:** Do not delete the rows for unused team members.

3. Using an online random number generator, calculate the base number of points that each member of the team will deliver during the sprint. Each team member will deliver between 10 and 20 story points per sprint. This isn't a reflection on you personally so don't be competitive about it – the numbers don't matter, they just make the game a bit more interesting and different from team to team.

**Note:** Do not use the random number generator within Google sheets.

4. Sum up the total number of story points that you estimated for all of your stories and place it in cell B5.

### 9.3 The Game

During sprints 2 and 3, you will play the game for 6 sprints of your project. Each sprint of the game is done the same way.

Each round of the game requires a random number generator to generate integers between 1 and 20. In the worksheet this is referred to as “rolling a d20.” You can blame your instructor’s long familiarity with Advanced Dungeons & Dragons.

**Note:** You will record all of the moves in the game in the “Plan” sheet of the worksheet. However, you will need to reference the “Productivity” and “Scope” sheets for each sprint.

#### 9.3.1 Sprint Planning

At the start of each sprint, your team must decide which stories you will be able to complete.

**Note:** For the first sprint this will be equal to the “Total Team Points” for sprint 0. For subsequent sprints this will be the number of points you completed in the previous sprint.

Add stories from the product backlog into the Trello list for the current sprint. The Trello template contains a list for the Sprint 1 Backlog already. Create the other lists as needed.

**Note:** If you brought stories forward from the previous sprint, DO NOT reduce the estimate. Stories are either completed in a sprint or not – there is no partial credit. If you want to demonstrate progress on larger stories, make sure you decompose them into separate stories with smaller estimates that are more likely to be completed within the sprint.

#### 9.3.2 Determine Sprint Productivity

Each sprint can be impacted by factors outside of the team’s control. There is some natural variability to our work, but there are also factors including: time out of the office such as sick time or vacation; unplanned work like production outages; and non-project work such as interviews, training, mentoring and the like.

At the start of each sprint, each team member should roll one d20 for their personal variability. This will affect how many story points of work you can deliver during the sprint. Place the value corresponding to your d20 roll next to your name in the column for that sprint. That value will be added to your base story points within the rest of the sheet.

Next, the team should roll three d20 values and capture the associated values from the tables for “out of office”, “unplanned work”, and “non-project work.” Record these values in the corresponding sections for those categories for that sprint.

#### 9.3.3 Updating the Sprint Scope

During each sprint you will do more than complete work. You will also identify new stories, re-estimate existing stories, and remove stories that are no longer relevant.

As a team, you will roll three d20s: one for added stories, one for changed estimates, and the third for removed stories.

**Added Stories** The first roll determines how many new stories will be added to the scope of the project. New stories can manifest in two ways. First, they can be brand new, never before seen stories. Second, they can be reflected as stories that had been previously removed from the project that the manager would now like re-added.

Record the number of story points of added stories on the worksheet in the section “ACTUAL > ...from Added Stories” for the sprint. Next, add that many points of stories to the project backlog. If you have any stories in your “Removed” list in Trello, add that many points worth of them back to your project backlog. If you do not have enough removed story points in your “Removed” list, then create placeholder stories. These are stories that are written to reflect the fact that some work is going to be done – you don’t need to actually come up with actual project-oriented stories for these. I’d suggest a naming convention for the story to reflect when it was added (e.g. PLACEHOLDER S1 STORY A). Remember that we’re still following the power-of-two for our estimation scheme, so you may need to create multiple placeholder stories to get the numbers right.

**Changed Estimates** The second roll determines how much re-estimation you need to do for stories in upcoming sprints.

Record the number of story points of changed estimates on the worksheet in the section “ACTUAL > ...from Changed Stories” for the sprint. Next, update some of the stories in your upcoming sprints to reflect that total change in story points. It doesn’t matter which stories you change, but make sure your updated estimates remains consistent with the powers-of-two scheme that we’ve adopted.

**Removed Stories** The third and final roll determines how many stories will be removed from the scope of the project. Removed stories can come either from placeholder stories or, if there are no placeholder stories, from real stories in your various upcoming sprints.

Record the number of story points of removed stories on the worksheet in the section “ACTUAL > ...from Removed Stories” for the sprint. Next, remove that many points of stories from the various sprints and place them into the “Removed” list within your Trello board. If you have any placeholder stories in your project backlog, remove that many points worth of them. If you do not have enough placeholder story points, then remove some actual stories.

### 9.3.4 Wrapping Up the Sprint

At the end of the sprint, you will have a total number of completed points. Move that many points worth of stories to the “Completed” Trello list. You may only move whole stories so must either have completed all of the points in the story or none may be counted. For any stories not completed in the sprint, you must decide whether or not they should be moved to the next sprint, some other sprint, or returned to the product backlog.



## 10 Peer Evaluation

50% of your project score is based on the team's performance. The other 50% of the project score is based on your team's evaluation of your contributions. At the end of the term, you will be asked to submit a peer evaluation of each of your teammates. This evaluation is intended to reflect your peers' understanding and ability to apply the agile practices and techniques that make up this project.

The peer evaluation determines how many of the project contribution points you will receive for your work during the term. Not everyone contributes equally to a project and thus not everyone should earn the same rewards for its successful completion. Your total peer evaluation score will be based on the median (middle value) of the peer evaluations provided by your team, including your own.

Please fill out the form on the following page for yourself and each of your team members and submit it as a PDF to D2L for the "Peer Evaluation" assignment once it becomes available. Guidance on the evaluation criteria and scale is provided on the next page to help you in your assessment. You may provide additional comments below the table if you wish, but only your scores in the table will be used in the score.

If you do not submit a peer evaluation form, then you will receive a "no show" (0%) for your self-evaluation, while each of your team members will receive "excellent" (100%) for your evaluations of them. Late evaluations will not be accepted.

### Example

Suppose that your team receives full marks for each of the 4 sprints. There are 125 points per sprint for a total of 500 team points earned by each member of the team.

Suppose further that your median contribution is evaluated at "Average." You would then receive 75% of the team points for your project contribution score. You would thus receive a total of 875 points for the project (500 team + 375 contribution). If another individual on your team did not contribute as significantly and received a median evaluation of "deficient" (25%), then they would receive 25% of the project's 500 points for their contribution score, yielding a total of 625 points (500 team + 125 contribution) for the project.

**Team Member Name:**

DEPENDABILITY	Always	Usually	Sometimes	Rarely	Never
Comes to our meetings prepared.	5	4	3	2	1
Does the assigned tasks well.	5	4	3	2	1
Does the assigned tasks on time.	5	4	3	2	1
Makes timely responses to project communications.	5	4	3	2	1

TEAM PLAYER	Always	Usually	Sometimes	Rarely	Never
Seems invested in the team doing well.	5	4	3	2	1
Exhibits friendly professionalism.	5	4	3	2	1
Is respectful of others.	5	4	3	2	1
Did a fair share of the work.	5	4	3	2	1

PROJECT PARTICIPANT	Always	Usually	Sometimes	Rarely	Never
Proposes (but doesn't force) new ideas, suggestions, courses of action.	5	4	3	2	1
Builds on or extends others' proposals.	5	4	3	2	1
Expresses support for other team members' opinions or ideas.	5	4	3	2	1
Disagreement with other team members' opinions or ideas is done respectfully.	5	4	3	2	1
Accepts or respectfully negotiates when other team members disagree with his/her ideas.	5	4	3	2	1
Invites views or opinions from team members not actively participating in the discussion.	5	4	3	2	1

Based on your responses to the above questions, assign an overall rating from the choices below that you think is the fairest assessment of the team member's contribution to the overall project.

Excellent	Very Good	Satisfactory	Average	Marginal	Deficient	Unsatisfactory	Superficial	No Show
100%	95%	85%	75%	50%	25%	15%	5%	0%

## 10.1 Scale

Each criteria should be evaluated on a scale of 0 to 5, with 5 being the highest score and 0 being the worst. Please use the following descriptions of each level as guidance.

<b>Excellent</b>	Consistently carried more than his/her fair share of the workload. They rarely relied on others and were often an individual on whom others relied.
<b>Very Good</b>	Consistently did what he/she was supposed to do, very well prepared and cooperative. Demonstrated a high degree of skill. Made few minor mistakes. Occasionally relied on others, but consistently contributed to the success of their peers.
<b>Satisfactory</b>	Usually did what he/she was supposed to do, acceptably prepared and cooperative. Demonstrated reasonable competence. They sometimes made mistakes, but recovered from them, learned from them, and did not make the same mistake again. Relied on others at the beginning, but by the end of the project were contributing as an equal.
<b>Average</b>	Often did what he/she was supposed to do, minimally prepared and cooperative. Did not volunteer for additional activities or to assist their team members who might have been over-committed.
<b>Marginal</b>	Sometimes failed to show up or complete assignments, rarely prepared. Did not demonstrate competence. Made frequent mistakes, sometimes learning from them, sometimes not. Relied heavily on others at the beginning but were able to gradually improve their own skills although they never contributed an equal share to the success of the project.
<b>Deficient</b>	Often failed to show up or complete assignments, rarely prepared. They often made mistakes but did not learn from them. They relied heavily on others and did not try to improve their own skills.
<b>Unsatisfactory</b>	Consistently failed to show up or complete assignments, unprepared. They were disruptive to the team. They made mistakes and failed to learn from them assuming they even tried at all. They relied heavily on others and made no attempt to improve their own skills.
<b>Superficial</b>	Practically no participation. Shows up at the last minute looking to help in the hopes of improving their peer evaluation score.
<b>No Show</b>	No participation at all.

## 11 Frequently Asked Questions

I've had some recurring questions about the group project for the class. I wanted to pull together some these questions and my responses to them.

---

**Q:** You said in class that the classroom isn't like the real world. Then why are you making us do a group project?

**A:** I did indeed say that. However, that doesn't mean that we can't try to inject some realism into the course. Meaningful software development rarely happens as a solo project. The techniques you're going to learn are only meaningful when viewed through the lens of group-based software development and they will remain utterly abstract unless you get the chance to practice them in a team environment.

---

**Q:** Why should my grade depend on other people who might not have any work ethic?

**A:** This one comes up during every group project and it reflects an arrogance that has no place in the classroom or the workplace. If shouldn't your grade depend on other people? Contrary to the narrative many of us have grown up with, your success in most areas of endeavor is not due solely to your own efforts. We're each constantly receiving help from others. A wise person recognizes that and does not take more credit than is their due. An even wiser person will help others to be their best. This argument, however, is one of the reasons that you have the ability to provide a peer evaluation at the end of the term. That's your best means of holding your group mates accountable to their commitments.

---

**Q:** Why are you making it harder to collaborate by including DL students in the in-class groups and vice versa?

**A:** It's good experience for folks who haven't worked with a distributed group to get a chance to do so. You'll find that some of what we discuss in class needs to be adjusted once everyone isn't physically co-located. Look at this as an opportunity to practice and hone these skills since it's becoming more and more common in our industry.