# SOFTWARE ENGINEERING DESIGN COLLABORATIVE ACTIVITY

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Engineer</td>
<td>• Focus on software engineering design process</td>
<td>• Completes Engineering Design Packet</td>
</tr>
<tr>
<td></td>
<td>• Primary decision-maker</td>
<td>• Delivers presentation to audience (or assigns designee)</td>
</tr>
<tr>
<td></td>
<td>• Prepares final report</td>
<td></td>
</tr>
<tr>
<td>Computer Scientist</td>
<td>• Focus on software program</td>
<td>• Inputs program codes</td>
</tr>
<tr>
<td></td>
<td>• Codes program</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Tests and adjusts programming</td>
<td></td>
</tr>
<tr>
<td>Computer Engineer</td>
<td>• Focus on hardware-software interface</td>
<td>• Identifies hardware/software constraints for final report</td>
</tr>
<tr>
<td></td>
<td>• Identifies hardware/software constraints, including devices/browsers</td>
<td>• Posts product on Web, composed text for audience</td>
</tr>
<tr>
<td></td>
<td>• Creates video</td>
<td>• Creates multimedia (video/ppt)</td>
</tr>
<tr>
<td>Technical Writer</td>
<td>• Focus on software program</td>
<td>• Creates flow chart for audience</td>
</tr>
<tr>
<td></td>
<td>• Prepares visuals and flow chart</td>
<td>• Creates additional videos (pictures, drawings, etc)</td>
</tr>
<tr>
<td></td>
<td>• Documents development with visuals</td>
<td></td>
</tr>
</tbody>
</table>

## REQUIREMENTS

<table>
<thead>
<tr>
<th>REQUIREMENTS</th>
<th>PTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTIMEDIA: Title and names of group members (Software Engineer)</td>
<td></td>
</tr>
<tr>
<td>MULTIMEDA: Illustrations/photos of software features (Technical Writer)</td>
<td>25</td>
</tr>
<tr>
<td>WEB: Directions/access to software program via multiple devices (Computer Engineer)</td>
<td>25</td>
</tr>
<tr>
<td>DEMONSTRATION: Show features of software program (Computer Scientist)</td>
<td>25</td>
</tr>
<tr>
<td>PRESENTATION: Reviews all information above (Software Engineer)</td>
<td>25</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
# SOFTWARE DESIGN ACTIVITY

## ASK?
The first step of the engineering design process is to ask. Engineers must define the problem and understand what to work on.

- Complete Step 1 in the NASA Design Packet.
- Define the problem.
  - Project Goal: Create an interactive animation of a word or phrase.
- Review the roles for each group member.
- Review examples of animated words and phrases at [https://scratch.mit.edu/studios/432299/](https://scratch.mit.edu/studios/432299/)

## IMAGINE
The second phase of the engineering design process is to imagine. Brainstorming allows you to come up with a wide range of ideas or solutions to the problem.

- Complete Steps 2/3 in the NASA Design Packet.
- Brainstorm what you want your interactive animation to include.
- Identify constraints on your project (most specifically, you can only use Scratch® and you have a limited amount of time)

## PLAN
The third phase of the engineering design process is to plan. Pick one promising idea and then develop a plan by sketching a flow chart of the design.

- Complete Step 4 in the NASA Design Packet.
- Choose the word/phrase you want to animate.
- Make a list of the animation elements you want to include in the order of action.
- Identify the background you’d like to have.
- Identify the interactive component of your animation.

## CREATE
The fourth phase of the engineering design process is to create. Emphasize teamwork and follow the plan.

- Complete Step 5 in the NASA Design Packet.
- Code your animation.
- Create an account at [https://scratch.mit.edu/](https://scratch.mit.edu/)
- Follow steps at: [http://scratch.mit.edu/name/](http://scratch.mit.edu/name/)
- Create a flowchart of your software program.

## EXPERIMENT
The fifth phase of the engineering design process is to experiment. Test one or more variables through experimentation. Use results to determine how to improve your design.

- Complete Step 6 in the NASA Design Packet.
- Test your program often. Measure and record data.
- Debug your code as needed.
- Revise your flow chart to reflect changes.
- Document your process with pictures and illustrations.

## IMPROVE
The final phase of the engineering design process is to improve. This is the step where changes are made to the original model to come up with the best solution.

- Complete Step 7 in the NASA Design Packet
- Test the complete program and adjust as needed.
- Ask your instructor, an observer, or someone from another group to review and critique.
- Compose instructions, notes, and credits for the public launch of your program.

## FINALIZE & SHARE
This extra step is to finalize your project and share the completed program with your virtual and/or physical audience.

- Complete Step 8 in the NASA Design Packet.
- Share your project by following directions in Scratch®
- Send the link to your program to your instructor.
- Plan and create your presentation – will you provide a virtual or FTF demonstration? Will you create a ppt or movie? How will you share the link to your program?
- Review the Rubric in the NASA Design Packet and revise any steps needed to insure full credit.
SOFTWARE CODING VOCABULARY

Terms in blue are linked to further information on the Scratch Wiki.

• Algorithm - a list of steps to finish a task. For example, the collection of steps to make a peanut butter and jelly sandwich is an algorithm.

• Animation – feature that includes moving images.

• Blocks — puzzle-piece shapes that are used to create code in Scratch. The blocks connect to each other like a jigsaw puzzle, where each data type (event, command, reported value, or script end) has its own shape and a specially shaped slot for it to be inserted into — this prevents syntax errors. Blocks make up scripts, which make a project work. The order of blocks are very important, as they determine how sprites interact with each other and the backdrop.
  o Cap Block — block which ends a script or project
  o Control Block — block that either starts a script or keeps a script running
  o Looks Block — block which controls how a sprite or the stage looks
  o Motion Block — block which controls a sprite's movement
  o Operator Block — block that performs math functions
  o Sensing Block — block that detects something, from a sound from a microphone to whether a sprite is touching a certain color
  o Sound Block — block which controls a sound
  o Variable Block — block that controls a variable

• Binary - way of representing information using only two options.
• Bug - error in a program that prevents the program from running as expected.
• Code - One or more commands or algorithm(s) designed to be carried out by a computer.
• Command - instruction for the computer; many commands put together make up an algorithm
• Conditionals - statements that only run under certain conditions or situations.
• Data - quantities, characters, or symbols that are the inputs and outputs of computer programs.
• Debugging - finding and fixing errors in programs.
• Default Value — default value of an insert in a block.
• Direction — where a sprite is pointing.
• Event - action that causes something to happen.
• Function - piece of code that you can easily call over and over again.
• Interactive — feature that is designed to involve the user.
• Loop - action of doing something over and over again.
• Program - algorithm that has been coded into something that can be run by a machine.
• Run program – to cause a computer to execute the commands you've written in your program.
• Toolbox - location of commands you use to write your program.
• Variable - placeholder for a piece of information that can change.
• Workspace - location of space where you drag and drop commands to build your program.
• Duplicate — to copy and create another.
• Script - collection or stack of blocks that all interlock with one another.
• Sprite — An object in Scratch that performs functions controlled by scripts.
• Scratch — A drag and drop programming language.
• Scratch 2.0 — The current version of Scratch.
• X Position — position of sprite along the horizontal axis.
• Y Position — position of sprite along the vertical axis.