1. 1. System A for unit testing - to validate the individual module
   2. System C for integration testing - to test multiple (individually proven) modules to the main system
   3. System B for system testing - to test the completed system as a whole
2. My system was a simple community posting site (twitter, tumbler). I had the User as a unit which would make posts of words, photos, videos, etc. owned by them held on a server, while other users comment on that post and vice versa.
   1. Verification
      1. Are the users account secure from others accidental login
      2. Is the site accessible from the browser
   2. Validation
      1. Do users want to put photos and videos in the same post?
      2. Do users want the ability to become anonymous while commenting?
   3. Functional
      1. Can posts be read?
      2. Can posts be created?
   4. Non-functional
      1. Can multiple people log into the site at the same time with different accounts?
      2. Are new posts being displayed without a manual refresh?
3. <https://github.com/AJ-Protzel/Leap-Year-Error-Handeling>
4. A restaurant owner wants a food delivery app for mobile. During this pandemic their business has been declining so the app must be made quick, easy to recognize being their restaurant, display their food, display their prices, display their delivery options.

| Testing Agile | Testing Waterfall |
| --- | --- |
| Create app framework, display food images - ask if this is the right style | Build app framework, display food and prices - make sure items are displayed clearly - internal testing |
| Add button functionality - click food, add to cart, checkout - ask if things like discounts or tip should be added | Add button functionality, make it clear with images, prices, total, add to cart, checkout |
|  | Present product to customer |