**CIS 305: Introduction to Software Development** Project Assignment 7: Introduction to Software Design

Getting Started

* There are multiple parts to this worksheet. Please **complete all parts** before submitting.
* Use the below listed articles, and any other online resources, to answer questions.
* To include multiple lines in your answer, press **SHIFT+ENTER**

Helpful Resources

* [How to Start a Software Development Project](https://www.sam-solutions.com/blog/how-to-start-a-software-development-project-2017/)
* [The What, Why, and How of Project Requirements](https://projectriskcoach.com/the-what-why-and-how-of-project-requirements/)
* [Requirements Engineering — Requirements Validation](https://medium.com/omarelgabrys-blog/requirements-engineering-requirements-validation-part-6-29778d7bde24)
* [Software Engineering | Requirements Validation Techniques](https://www.geeksforgeeks.org/software-engineering-requirements-validation-techniques/)
* [Sprint Planning Meeting Explained | Know all about Sprint Planning Meeting](https://www.youtube.com/watch?v=2A9rkiIcnVI)
* [The Sprint Backlog: Why It’s Important and How to Make it Great](https://www.sealights.io/sprint-velocity/the-sprint-backlog-why-its-important-and-how-to-make-it-great)
* [Planning Poker | Story Point Estimation in Agile | Agile Estimation Techniques](https://www.youtube.com/watch?v=TxSzo3lwwWQ)
* [Software Requirements Prioritization Techniques You Should Know](https://apiumhub.com/tech-blog-barcelona/software-requirements-prioritization-techniques/)
* [Agile in Practice: Prioritisation using MoSCoW](https://www.youtube.com/watch?v=QfZo9cxnQgY)
* [Software Design Strategies](https://www.youtube.com/watch?v=gDpKqVQ-Juw)
* [5 Key Parts to Problem Solving in the Software Development World](https://medium.com/@RobbCode/5-key-parts-to-problem-solving-in-the-software-development-world-f4436b788180)
* [How To Think And Problem Solve In Coding](https://www.youtube.com/watch?v=Hb9WUEXdkCE)
* [Don't Learn To Code In 2020 … (Learn to Problem Solve)](https://www.youtube.com/watch?v=GI_G3oGYLeo)
* [What’s an Algorithm?](https://www.youtube.com/watch?v=6hfOvs8pY1k)
* [Algorithm in Programming](https://www.educba.com/algorithm-in-programming/)
* [How to write a Pseudo Code?](https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/)
* [How to Use Flow Charts to Show how Processes Work](https://www.youtube.com/watch?v=ba5duMARdMc)
* [Pseudocode and Flowchart - Programming for beginners series](https://www.youtube.com/watch?v=AlEBMhxreo0)
* [Programming Basics: Creating an algorithm/flowchart and then adding a counter](https://www.youtube.com/watch?v=AjoSfSd-5Nw)
* [Explain Algorithm and Flowchart with Examples](https://www.edrawsoft.com/explain-algorithm-flowchart.html)
* [Usability Testing in Design – Why is it important?](https://uxdesign.cc/usability-testing-in-design-and-why-is-it-important-cfddfbbdaac9)

# Part 1. Requirements Gathering and Validation

1. List four things that a team should have to start a software development project.   
   Clearly defined requirements  
   Estimates to Sprint Backlogs  
   Design team  
   User Acceptance Testing Environment
2. List one difference between functional and non-functional requirements.  
   Functional requirements – describes the behaviors of the product   
   Non-Functional requirements – describes the environmental conditions or qualities required for the product to be effective
3. List four steps in the requirements phase.   
   Elicit, Analyze, Document, Validate
4. List the five different types of tests to check the requirements, during the requirements validation phase.  
   Validity, Consistency, Completeness, Realism, Verifiability
5. List three techniques to validate requirements.  
   Prototyping   
   Test case generation  
   Automated Consistency Analysis
6. List three things to check, when reviewing requirements being provided to your team, that helps identify ambiguity.  
   Checking early whether the idea is feasible or not.  
   Obtaining the opinions and suggestion of other people.   
   Checking the approval of others and reaching an agreement.

# Part 2. Planning and Prioritizing with a Team

1. List two things the sprint planning meeting focuses on.  
   Selecting Product Backlog Items  
   What is to be built and how the team will build it
2. List one reason why breaking backlog items into tasks is done by the development team, during spring planning.  
   helps confirm whether the right amount of work is tasked
3. List two provided tips for making your spring backlog effective.  
   prioritized each task according to the needs of the product owner.  
   Team members should assign tasks. They should revise their initial commitment to sprint scope. The backlog should be dynamic and updated daily
4. In 1-2 sentences, describe why prioritization of requirements is important for a software development project.  
   Software requirements prioritization helps the project manager resolve conflicts, plan for staged deliveries, and make the necessary trade-off decisions
5. List two reasons why prioritization can be challenging for a team.  
   Pressure of trying to accomplish a great deal with limited time and limited resources.  
   Once priorities are determined only the Product Owner can make revisions.
6. List two popular software requirements prioritization techniques.   
   Planning Poker*,* Dependency map, MoSCoW
7. In the MoSCoW method for prioritization, name each of the categories.   
   MUST – mandatory SHOULD – of high priority COULD – preferred but not necessary WOULD – can be postponed and suggested for future execution
8. How are estimates of development tasks usually expressed? For example, the maturity of pets are usually expressed in age, while the popularity of a social media personality is usually expressed in number of followers.  
   Estimates are expressed in an added total value metric with ratings 1-9 with 1 being the least useful and 9 being most impactful to evaluate: relative benefit that each feature provides to the customer, relative penalty the customer or business would suffer if the feature is not included, cost of implementing each feature, degree of technical or other risk associated with each feature
9. In 1-2 sentences, describe the purpose of using the planning poker technique in agile development.  
   Planning poker is a consensus-based technique to provide a quantified evaluation to obtain an indication of duration and effort needed for the product backlog

# Part 3. Software Design

1. List the three common approaches to approaching software design.  
   Top down  
   Bottom Up – merge pre-existing modules  
   Object Oriented – Abstraction, inheritance, polymorphism (data focused)
2. In the How to Think and Problem Solve in Coding video, list the three steps he prescribes in problem solving.  
   Understand and describe the problem in simplicity  
   Break problem down into component parts  
   Plan of Action
3. In the Don't Learn to Code in 2020 video, what does the speak believe the main benefit of breaking the problem down is.  
   The main benefit is problem solving because code syntax can be looked up and translated
4. In 1-2 sentences, describe what an algorithm is.  
   specific and logical procedure to be followed in order to achieve specific results, or to solve a math problem
5. In 1-2 sentences, describe what pseudocode is.  
   English like syntax that resembles a programming language. Programming description that does not require any strict programming language syntax or underlying technology considerations
6. In 1-2 sentences, describe what a flow chart is used for.  
   Flow charts show step-by-step a process or task, certain actions represented by symbols so the observer can understand what to do at each stage
7. List the name of the 3 main shapes used in flow charts.  
   Elongated circle  
   rectangle  
   diamond
8. List 1 reason why user testing is important to the software design process.  
   Usability testing answers the questions whether the product is functional, intuitive, engaging, error tolerant and valuable
9. List 1 reason why it is helpful to test for specific requirements, when conducting user testing.  
   Conducting user testing can increase ROI, provide immediate feedback from users on how to enhance the experience and what features need added or removed.