## Project Management

To develop software, a software development process is a division of software development work into specific stages including activities to satisfy better planning and management. The main target of this process is delivering software to clients in a constrained time. Sometimes, this methodology consists of defining a package created and completed by a team to produce or develop an application [1].

While a specific software has grown in complexity, the programming task requires a team effort. A team-based project needs project management to provide a workflow with milestones and deliverables to distribute tasks, effective communication and report progress to prepare a final product [2].

The first step of developing software is defining a life cycle for software development which is similar to an assembly line in manufacturing. In the 1960s and 1970s, many of the businesses that moved into software engineering were manufacturing industries, resulting in reinforcing this philosophy. The idea of considering software as a product has come from those years because industries dealt with the final step of the assembly line named “product”. One of the common characteristics of these systems was that the user's requirements were quite static [3].

### Waterfall model

One of the oldest models for the software life cycle is the “waterfall model”. This model was introduced to identify standard sequences of activities in the creation of software. This standard is similar to manufacturing a product in industries. Value-adding and feedback steps are two vital elements of the waterfall model. In another word, a rigid phase uses when one step ends, the next starts. Steps occur in a sequence way that indicates this model does not allow developers to go back to previous steps. It is called the waterfall because the water flows from up to down and it cannot go back up [2].

Waterfall has the following characteristics:

- Separate phase: the model has well-defined and discrete steps that can be managed differently by different team members.

- Requirement phase: in this stage, the whole system is well defined. The research operates on the environment of users and questioning of the user to understands the needs of the project. Due to the rigid structure of this model, each activity should be complete and usable which is difficult.
- Architecture phase: in this part general system design has been investigated to fulfill the requirements while operating within the contains of the target production.
- Detailed design phase: High-level design is considered in this step. High-level design is a breakdown into subsystem and module. At this point, internal design how-know tends to predominate over the user and operational environment knowledge, although the design must (should) support the requirements at all times.
- Code: In this step, all know-how from the previous phases is converted to executable form. Also, some additional knowledge like language capabilities and limitations or target processor operation is added at this point.
- Unit test: This is the first feedback stage. the result of the coded module is compared to a set of expected outputs. Any difference is studied and find the root of this difference.
- System test: software consists of several functions and classes. The interconnectivity of each of these elements should be examined as a unified system. The modules are aggregated in increasing size, quantity, and complexity, and a variety of testing functions are performed against the combinations. Furthermore, there may also be load tests to make sure the system loads in the varieties of user target environments; acceptance tests run to certify the system; performance tests to prove performance and capacity levels.

Despite the upside of the waterfall model, there is some disadvantage of this model that:

- It is often difficult to get complete requirements in a project, because business people have not really thought in detail about what they need, and requirements can change during the project.
- This model needed a detailed breakdown of the task and deliverables which is really difficult to distributed tasks based on the team’s capability.

- waterfall projects do not inherently have to span long periods of time, but it is very common for these projects to span months or quarters because of the emphasis on trying to get everything done at one time. The possibility of projects being late, over budget, and failing to meet expectations rises as the timeframe for an IT project significantly increases [4].

**Agile Model**

The agile model evolved as a reaction to rigid software development models like the Waterfall Model. Agile processes are fast, very iterative, and responsive to change. This makes sense because the word “agile” means responsive to change [5].

Agile is an iterative team-based method where requirements and deliverables evolve through cross-functional collaboration. Instead of creating schedules with detailed tasks, a list of deliverables is defined, with the team gathering requirements and creating the deliverables in time-boxed phases called sprints [4]. Each sprint has a target of what is to be built, a design and flexible plan that will lead to building it, the work, and the resultant product increment.” A sprint is the basic unit of work being done in an agile model. by the time, the complexity of system development will smaller and less complex systems. In the agile model, the chunking of tasks into a small scope for each iteration of the process was the vital element to getting the feedback necessary to track changes in understanding that occurred with real usage [5].

The workflow of agile methodology is presented in the following:

- Prioritizing deliverable based on the consumer/client request
- Gathering requirements, design, development, and testing in each sprint
- Reviewing and evaluating work by consumers
- Uncompleted works are reprioritized for the following sprints

In spite of the benefits of agile projects, this methodology has the following downsides:

- The deliverable should be relapsed rapidly which means sometimes scope might be reduced and quality of deliverables compromised.

- The problem of the wide variance in expectations about project costs, overall functionality, and quality can happen when the project is large [4].

**Selecting proper methodology:**

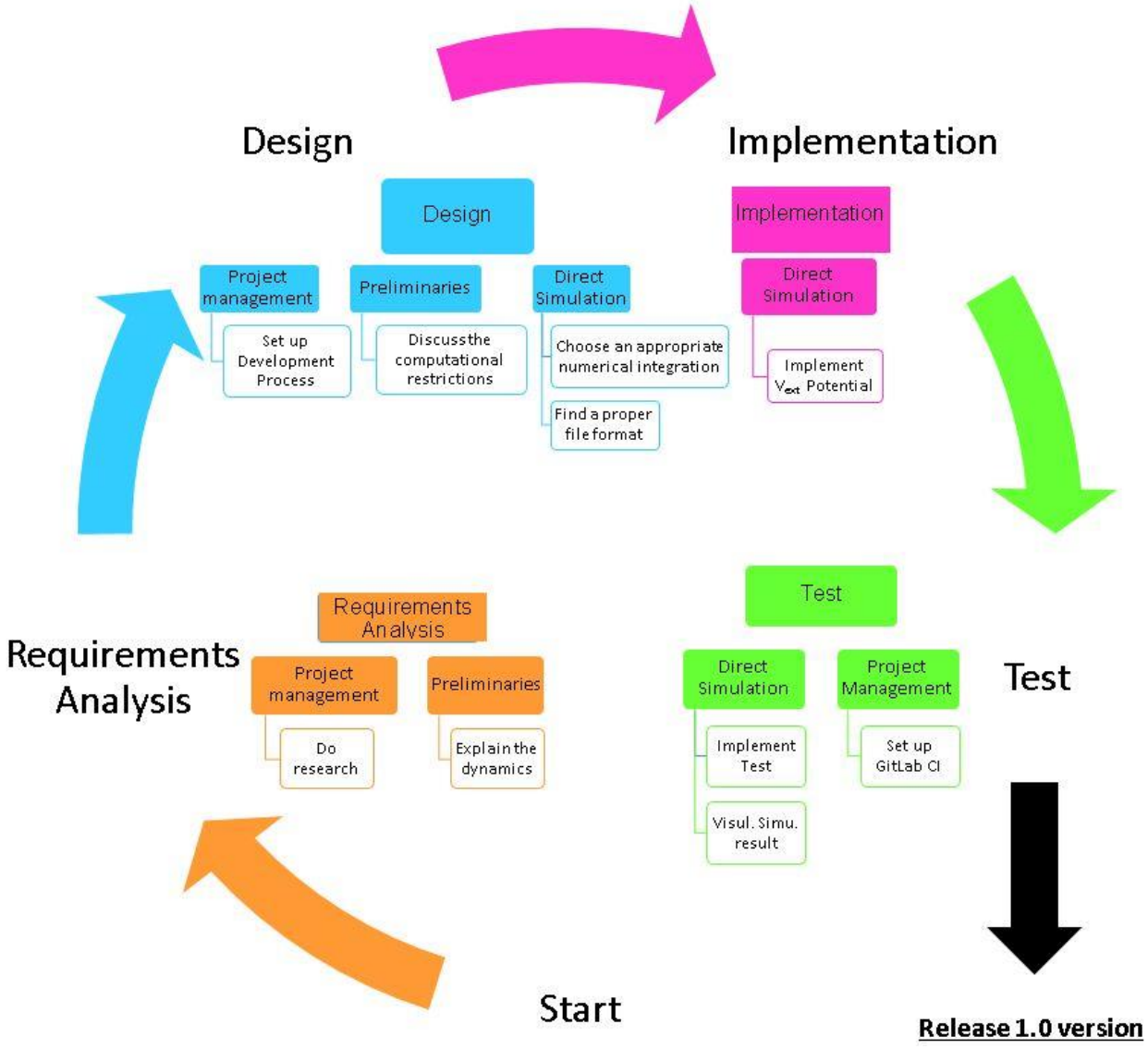
This is an important step of each software development project because it influences all activities during the project. There are several factors that developers should take into account when looking at methodology. Molecular Dynamics (MD) project is an interdisciplinary work that needs a wide range of specialists such as chemists, mathematicians, and software developers. To exchanges ideas in this environment, the team should communicate more frequently. Regular meetings and releasing deliverables rapidly can help them to recognize their issues, leading to rerouting their path to catch the team's goal. Instead of using the agile model in this multidisciplinary project, employing the waterfall model can isolate our group members to do a specific task without exchanging ideas. Another reason why this project should be managed by the agile model is that existing an end-user during the development process can be beneficial because it can be a potential to respond in each sprint which means group member can revise their plan and tasks immediately. Finally, project structure fits the agile model because the main challenge is calculating potential energy which is divided into three terms. Therefore, by creating a general structure in the first phase, team members can add other terms in the following stages.

**Molecular Dynamics Project Workflow**

In this project, the decision was made to develop a python software package in four sprints. mandatory and easy tasks are our prioritizing factors and most of the optional tasks will be implemented in the last sprint.

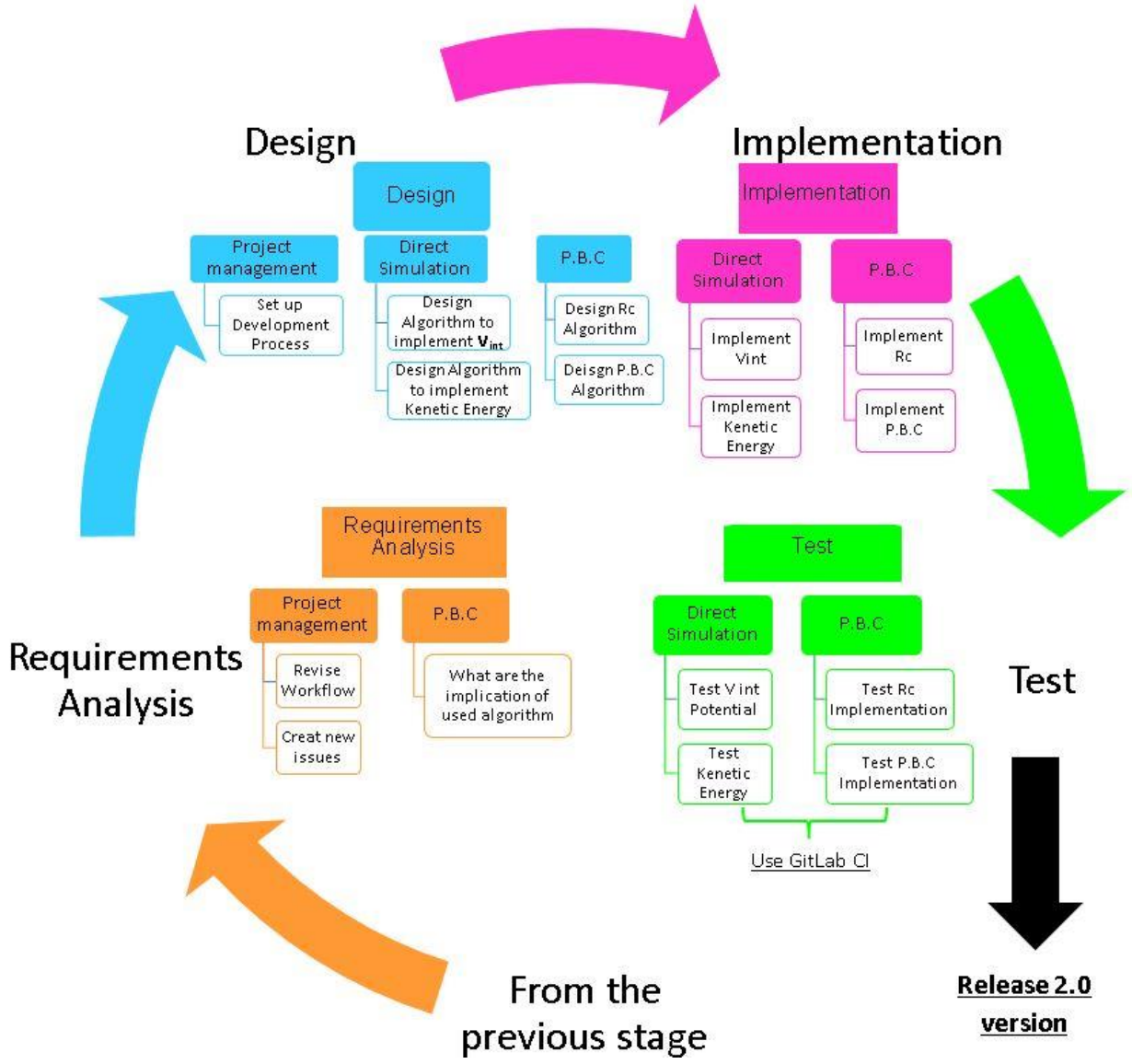
**First sprint:**

In tandem with developing workflow and selecting the development process, the dynamics of the system are studied in this stage. Also, computational cost and choosing the accurate numerical integrator is selected. Drawing molecular trajectories need a proper file format to store data and use the position of particles while visualization. Approximating and testing external potential and visualization is the target of this step.




**Second sprint:**

Firstly, the implication of the previous phase is studied and reported. modeling Internal potential and kinetic energy of water molecules are designed. Due to the computational cost of short-range interaction, the cut-off radius is considered to reduce the complexity of the model. Also, periodic boundary conditions can decrease the complexity of the model, so this concept is added to the simulation. In the following, code implementation and test simulation are done.




[1] https://courses.lumenlearning.com/zeliite115/chapter/reading-software-development-process/

[2] Conrad, Eric, Seth Misenar, and Joshua Feldman. Eleventh Hour CISSP®: Study Guide. Syngress, 2016.

[3] Armour, Phillip G. "The laws of software process." Commun. ACM 44.1 (2001): 1517.

[4] Sherman, Rick. Business intelligence guidebook: From data integration to analytics. Newnes, 2014.

[5] Hartson, Rex, et al. "Agile Lifecycle Processes and the Funnel Model of Agile UX." The UX Book (2019): 63-80.