Software Life Cycle

Software is a set of instructions, data or programs used to operate computers and execute specific tasks, so, software is an ordered set of activities; that produce a certain output expected.

A software process must specify software project management control, named as life cycle. SDLC (Software development life cycle) is a methodology that defines the entire procedure of software development step-by-step.

The SDLC key steps are:

1. Planning. Defining the project scope, goals, timelines, and resource requirements.
2. Analysis. Defining the system architecture and specifications.
3. Design. Creating a blueprint for the software system, outlining its structure and components. Defining the user interface, databases, and other technical specifications.
4. Implementation. Writing the actual code based on the design specifications.
5. Testing. Conducting various testing phases, including unit testing, integration testing, and system testing.
6. Deployment. Releasing the software to users or the production environment.
7. Maintenance. Addressing issues and bugs that arise post-deployment, and implementing updates.

As we know, Farah and Alexandru have provided information on the Alice and Atlas projects, we know that one of the functions of the experiments at the LHC is particle collisions.

So, Collisions of heavy ions produce a large number of particles in their final state, this represents a real challenge for analysis and reconstruction algorithms. The design of the detectors and the development of these algorithms require precise and predictive simulations of their response.

ALICE has designed a data acquisition system (DAQ) that operates satisfactorily during LHC collisions (protons and heavy ions). Additionally, ALICE's DAQ balances its recording ability central collision events and events with effective sections of rare events. For this task, this system has the capacity to process 2.5 Gb/second of information and to record 1.25 Gb/second

All Software (simulation environment) used in ALICE is known as AliRoot, which has the root environment as its platform. In general, AliRoot is built based on classes, which are responsible for carrying out specific tasks for one of the 2 main actions. Simulations and reconstructions of events.

The life cycle of the software can be explained if we analyze each task.

A. Simulation environment Life Cycle

The different types of simulation carried out in ALICE are: primary collisions (protons and heavy ions), generation of particles, transport of these through the detector, simulation of the energy deposited in each of them, the response of the detectors and the digital signal generation. They use AliSimulation Class. The processes involved are:

1. The generation of events is carried out using some generator.

2. Based on the previous generation, the simulated particles are transported to the different detectors considered in the simulation.

3. The energy deposited by the particles is recorded throughout their passage through the different detectors considered in the simulation.

4. The response of the detector to the passage of particles is recorded (summable digits).

5. A compendium is made of the data generated in the previous processes by event generated.

6. This information is recorded in objects known as digits or raw digits.

B. Reconstruction of events

After finishing the simulation of events in ALICE, we proceed to the reconstruction of them using files that contain digits or RAW data types.

1. Cluster the simulation data: identify sets of adjacent digits in space and/or time that were possibly generated by the same particle that crossed over some sensitive element of the detector.

2. Identify the different trajectories: carried out with a set of five parameters (as curvature and angles with respect to the origin of coordinates) of the trajectory of the particle together with the corresponding covariance matrix estimated at a point given in space.

3. Identify the particles coming from the generation.

4. Store all the information in a summary file (ESD6).
   1. Among the objectives of the reconstruction we have:
   2. Provide the necessary data for the physical analysis of a process.
   3. Have high efficiency and quality in data
   4. Have a simple interface to access the information contained in the ESD files.
   5. Have reconstructions with only a few detectors. This using The AliReconstructor base class, to define the interface to the reconstruction-

specific code for each detector. For this reason, each detector must develop a code for this task.

In ALICE, they work through 2 important ways, they analyze events through simulations and also events obtained with real data, the life cycle describe the tasks included.

Moreover, software must comply with certain requirements to accurate its quality control, and quality management processes, ISO 12207 is the software life cycle processes standard, so, The objective of this standard is to establish a global standard for the software life cycle, providing minimum requirements and best practices to ensure quality and efficiency in software development.

Certification to ISO 12207 demonstrates that an organization meets the requirements and best practices established in the standard, which increases its credibility and confidence in software development. ISO 12207 provides a framework for software life cycle processes, but it doesn't prescribe specific requirements for individual software systems like AliRoot.

How ISO/IEC 12207 principles could be applied to define requirements for AliRoot? ALICE has the document with demonstrates that AliRoot is able to execute the tasks in a correct way, fulfilling with requirements, this is an example, because the document is private so I don’t know about that specific information for AliRoot

Documentation			
Step	**Name**	**Requirement**	**Justification**
1	Acquisition Process:	AliRoot shall provide documentation specifying its functionality, interfaces, and system requirements for potential users.	This requirement ensures that users can understand the capabilities and limitations of AliRoot before acquiring and using it for their experiments.
2	Supply Process:	AliRoot shall be delivered with installation instructions, user guides, and release notes.	Providing comprehensive documentation and support materials facilitates the installation, configuration, and use of AliRoot by end-users.
3	Development Process:	AliRoot developers shall follow a documented software development process, including requirements analysis, design, implementation, testing, and maintenance.	Adhering to a structured development process ensures the reliability, maintainability, and extensibility of AliRoot throughout its life cycle.
4	Operation Process:	AliRoot shall support compatibility with various computing platforms commonly used in high-energy physics research.	Supporting multiple computing platforms ensures that AliRoot can be deployed and used effectively in different research environments.
5	Maintenance Process:	AliRoot shall provide mechanisms for bug reporting, issue tracking, and software updates.	Continuous maintenance and support are essential for addressing software defects, improving performance, and adding new features over time.

Documentation			
Step	**Name**	**Requirement**	**Justification**
6	Configuration Management Process:	AliRoot shall implement version control and configuration management practices to track changes to source code, documentation, and other project artifacts.	Effective configuration management ensures the integrity and traceability of software artifacts throughout their life cycle.
7	Quality Assurance Process:	AliRoot shall undergo regular quality assurance activities, including code reviews, testing, and validation against experimental data.	Quality assurance activities help ensure that AliRoot meets specified requirements, performs reliably, and produces accurate results for data analysis.
8	Documentation Management Process:	AliRoot shall maintain up-to-date documentation describing its architecture, algorithms, APIs, and usage guidelines.	Comprehensive documentation facilitates understanding, collaboration, and knowledge transfer among AliRoot developers and users.
9	Improvement Process:	AliRoot shall periodically assess its performance, usability, and user feedback to identify opportunities for enhancement and optimization.	Continuous improvement ensures that AliRoot remains relevant, efficient, and user-friendly in supporting high-energy physics research.