Requirements Gathering using Object-Oriented Models

Software Life Cycle

- The "software lifecycle" refers to all stages of software development from design to disappearance. The objective of such a division is to allow the definition of intermediate milestones allowing the validation of the software development, that is to say the conformity of the software with the needs expressed, and the verification of the development process. It means the adequacy of the methods used.

- The origin of this division comes from the fact that the errors have a higher cost because they are detected late in the process. The life cycle makes it possible to detect the errors at the earliest and thus to control the quality of the software, the delays of its realization and the associated costs.
The 8 steps of software development

The software life cycle includes the following activities:

- Goals definition, consisting of defining the purpose of the project and its inclusion in a global strategy.

- Analysis of needs, collection and formalization of the needs of the applicant (the client) and all the constraints.

- Conception/Design, This is the development of the specifications of the general architecture of the software and precisely define each subset of the software.

- Coding (Implementation or programming), or translation into a programming language of the functionalities defined during design phases.
The 8 steps of software development

- Validation, that is, verification of the software's compliance with the initial specifications.
- Documentation, aimed at producing the information necessary for the use of the software and for further development.
- Production - End user
- Maintenance, including all corrective (corrective maintenance) and scalable (evolutionary maintenance) corrective actions on the software.

The sequence and presence of each of these activities in the life cycle depends on the choice of a life-cycle model between the client and the development team.

Models / processes of the development cycle

- In order to be able to have a common methodology between the customer and the development service company, life cycle models have been developed defining the stages of the development as well as the documents to be produced allowing to validate each one of the steps before proceeding to the next. At the end of each phase, the reviews are organized.
The waterfall life-cycle model was developed in 1966, then formalized around 1970. It defines sequential phases at the end of each of which documents are produced to verify compliance before proceeding to the next:

- Specification
- General Conception
- Verification
- Detailed Conception
- Verification
- Coding
- Unit Test
- Integration
- Integration Test
- Production
- Validation
- Maintenance

Relationship between each step and the next:
- Outputs of a specific step are the entries for the next step
- New ideas are incorporated at the stage level to form the new deliverables
V Model (linear)

- May be considered as an extension of the waterfall model
- The V-lifecycle model assumes that procedures for verifying software compliance to specifications must be developed at the conception stages.
- Instead of moving down in a linear way, the process steps are bent upwards after the coding phase

Spiral model (iterative and evolutionary)

- The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models
- Real Validation, not in documents
- Risk limitation at each iteration
- Partner customer: quick return on expectations
- Progressivity: No surprises or new requirements / needs in the run-up to delivery
  - Flexibility: Modification of specifications = new iteration
  - Maintenance = iteration form
Spiral model (iterative and evolutionary)

1. Determine objectives
2. Identify and resolve risks
3. Development and Test
4. Plan the next iteration

AGILE

- RAD (Rapid Application Development)
- Unified Process (UP)
- XP (eXtreme Programming)
AGILE Approach

- The “agile” development models aim to reduce the life cycle of the software (thus accelerating its development) by developing a minimal version and then integrating the functionalities by an iterative process based on a customer listening and tests throughout the Development cycle.
- The origin of agile models is related to the instability of the technological environment and the fact that the client is often unable to define his needs exhaustively from the beginning of the project.
- The approach is based on individuals and interactions rather than on processes and tools.
- Software development rather than exhaustive documentation
- Openness to change rather than a rigid plan
- The client is a full-fledged pilot of his project and gets a very early start-up

RAD Model (Rapid Application Development)

- The “rapid application development model” defined by James Martin consists of a short development cycle based on 3 phases (Requirements, Design and Construction) within an ideal period of 90 days and a maximum of 120 days.
UP Model (Unified Process)

- The Unified Process model is an iterative and incremental development process, which means that the project is split into very short phases at the end of each of which a new incremented version is delivered.
- UML modeling for the description of the software architecture (functional, software and physical) and the development of user cases to describe user needs and requirements.
The XP model defines a number of best practices for developing software in optimal conditions by placing the client at the heart of the development process, in close relationship with the client.

EXtreme Programming is based on the following concepts: Development teams work directly with the customer on very short cycles of one to two weeks maximum.

Deliveries of software versions intervene very early and at a high frequency to maximize the impact of user feedback.

The code is tested and cleaned throughout the development process.

SQA activities must be integrated into the cycle
- At the level of transitions from one phase to another.
- May be: Inspection or review
- Review is accompanied by an acceptance test at the level of the transition from the test phase to the installation
- First steps: prevention of defects
- Intermediate phases: elimination of defects
- Operational Phase: Limit Defects