Modeling and measuring a software engineering course software process:

1. Process experiments at University of Pisa

- **Objective:** The main objective of the course software process is to provide an opportunity for students to apply software engineering methods and tools in a practical setting.
- **Methodology:** The process model is designed with the following requirements: software engineering methods, such as object-oriented analysis and design, are easy to learn as they rely on few principles, but they can be applied to a wide range of projects.
- **Process Model:** The process model defined at University of Pisa was reused and this process model prescribes metrics and tools rather than computer science mathematics such as the Z or LOTOS formal languages.
- **Implementation:** The software process model was defined by a notation derived from data flow diagram as the process modeling languages developed by the research community (see Figure 1).

2. Process experiences at Polytechnic of Torino

- **Objective:** The process model was designed to encourage a combined use of software process modeling, metrics, and teaching.
- **Process Model:** The software process model was defined at Polytechnic of Torino for which there is automatic tool support.
- **Implementation:** The software process model was implemented as a manual. The software process model was defined by a notation derived from data flow diagram as the process modeling languages developed by the research community (see Figure 1).

3. Process metrics management

- **Objective:** The process metrics management was designed to improve the interaction among teacher and students and to provide a constant updating of documentation.
- **Process Metrics:** The process metrics management was designed with the following requirements: the process model enaction must be generic, as it does not depend on a specific project, nor it is integrated with CASE technology.
- **Implementation:** The process metrics management was designed with the following requirements: the process model enaction must be generic, as it does not depend on a specific project, nor it is integrated with CASE technology.

4. Process documentation and software process model

- **Objective:** The process documentation and software process model were designed to provide support for process model enaction, not they are integrated with CASE technology.
- **Process Documentation:** The process documentation and software process model were designed to provide support for process model enaction, not they are integrated with CASE technology.
- **Software Process Model:** The software process model was designed as a set of process models, each model defines a part of the overall process. Process models can be used to describe the software processes, but they are not sufficient to provide support for process model enaction, nor they are integrated with CASE technology.

5. Process modeling technologies

- **Objective:** The process modeling technologies were designed to describe the software processes, but they are not sufficient to provide support for process model enaction, nor they are integrated with CASE technology.
- **Process Models:** The process models were designed as a set of process models, each model defines a part of the overall process. Process models can be used to describe the software processes, but they are not sufficient to provide support for process model enaction, nor they are integrated with CASE technology.
- **Implementation:** The process models were designed as a set of process models, each model defines a part of the overall process. Process models can be used to describe the software processes, but they are not sufficient to provide support for process model enaction, nor they are integrated with CASE technology.