

# yPBL: project based learning methodology applied to software engineering

Ernesto Expósito

CNRS ; LAAS ; 7 av. du Colonel Roche, F-31077 Toulouse, FRANCE

INSA - Université de Toulouse

- **The yPBL is a learning methodology, based on the PBL model and inspired in software engineering processes [Exposito2010].**
- **yPBL is aimed at being deployed in the context of software engineering courses based on the construction of a real software system.**
- **The yPBL model is defined as a mapping between the roles and phases considered in PBL methods into the roles, iterations and phases considered in the "y" process.**

[Exposito2010]: yPBL methodology: a problem-based learning method applied to Software Engineering, IEEE Engineering Education 2010 – The Future of Global Learning in Engineering Education (EDUCON'2010)

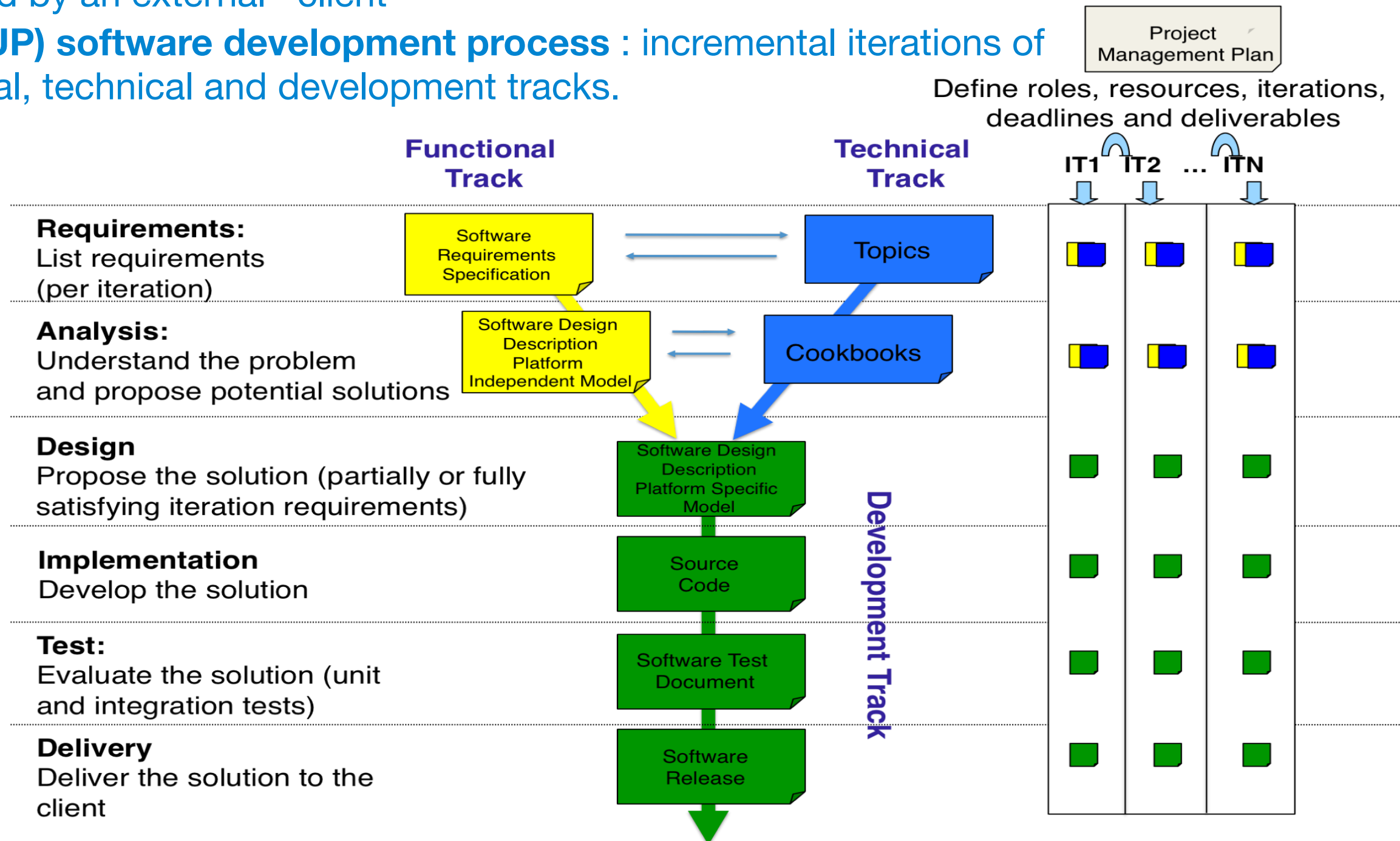
## Effective development of knowledge and skills in:

- **Software engineering:** project management, software development methodologies, requirements engineering, software architecture design and development, test validation and verification, quality insurance, etc.
- **Software technologies: distributed**
  - Architectures and styles : distributed / multi-tiers / web / multimedia / mobiles applications
  - Approaches : based on objects, design patterns, services, aspects, components
  - Research trends: semantic web (ontologies), autonomic / cloud computing, etc
- **Specialized English:**
  - Technical documents reading and writing
  - Projects and products presentations
  - Capacity of adapting language level to specialists and non-specialists

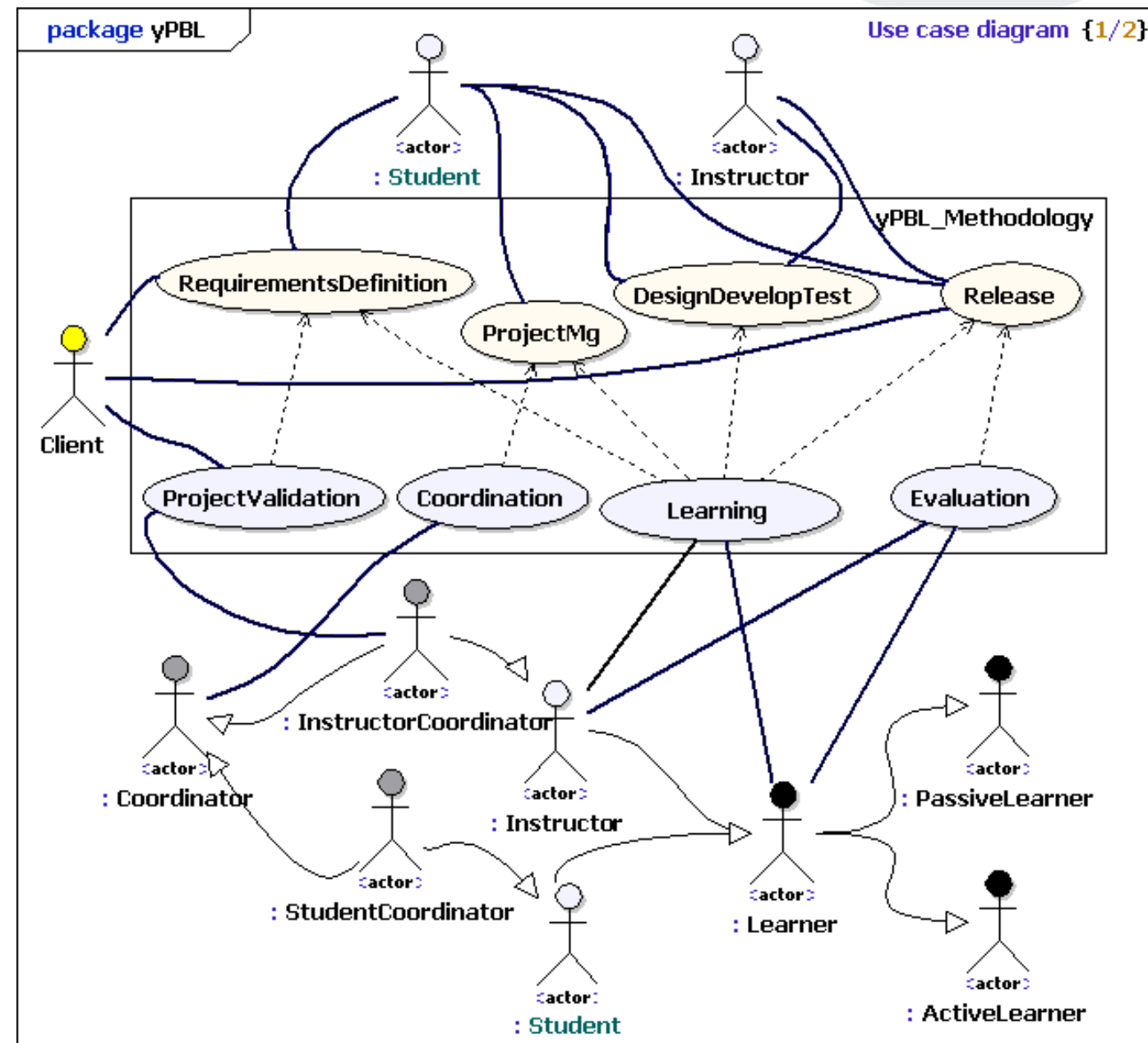
**Our observation:** Classical approach (courses, labs and exams) in mono-disciplinary mode is not well adapted to SE facets and technology evolutions

## Our approach : Combine

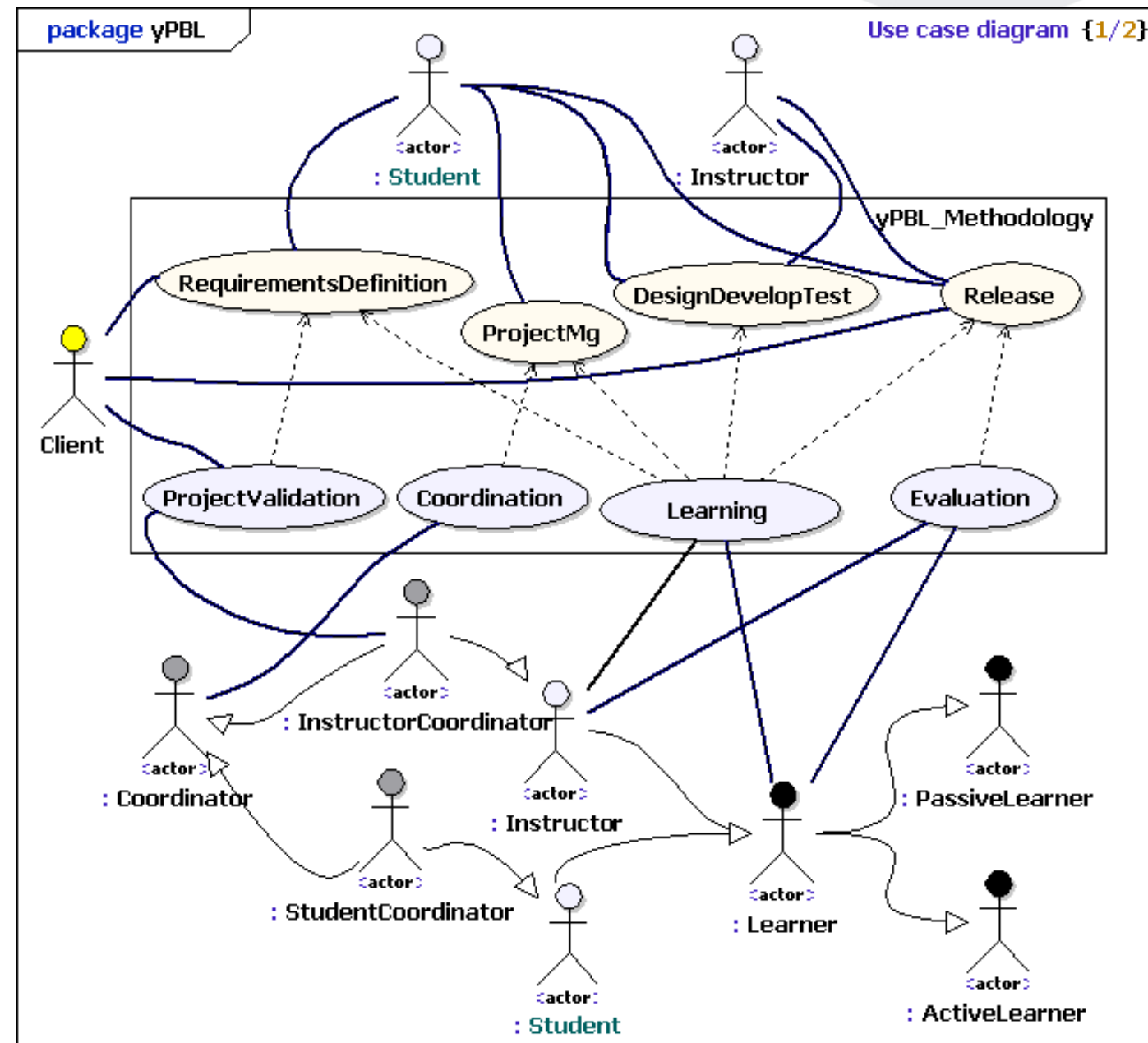
- **PBL (Problem Based Learning)** : learning based on a project proposed by an external 'client'
- **Y (2TUP) software development process** : incremental iterations of functional, technical and development tracks.



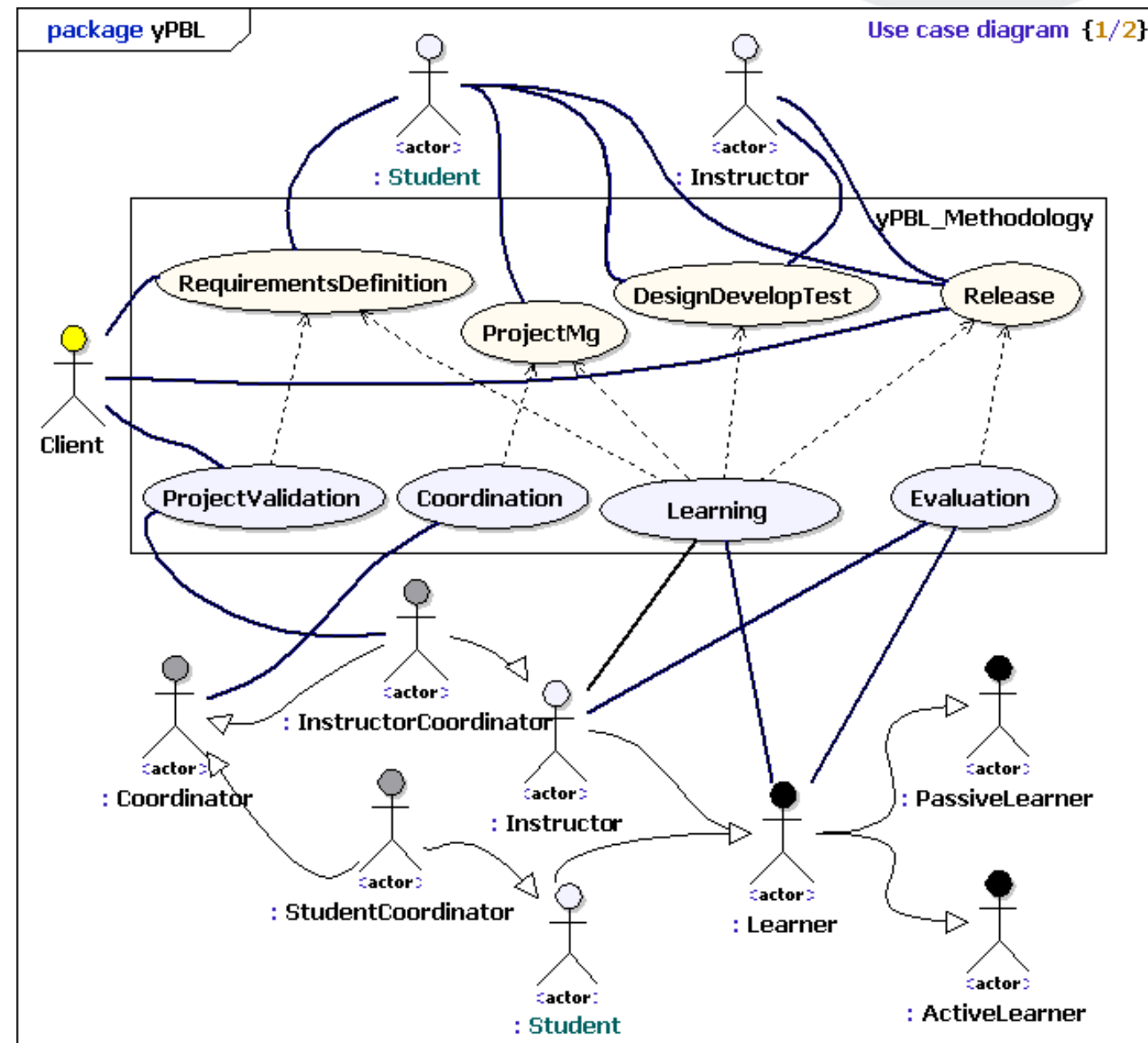
- In yPBL, the actors interacting to achieve the learning objectives while constructing a software system are the following: **students, instructors and the client.**
- Guided by the construction of the software project, two generalizations of actor roles are proposed in yPBL: **coordinators and learners.**



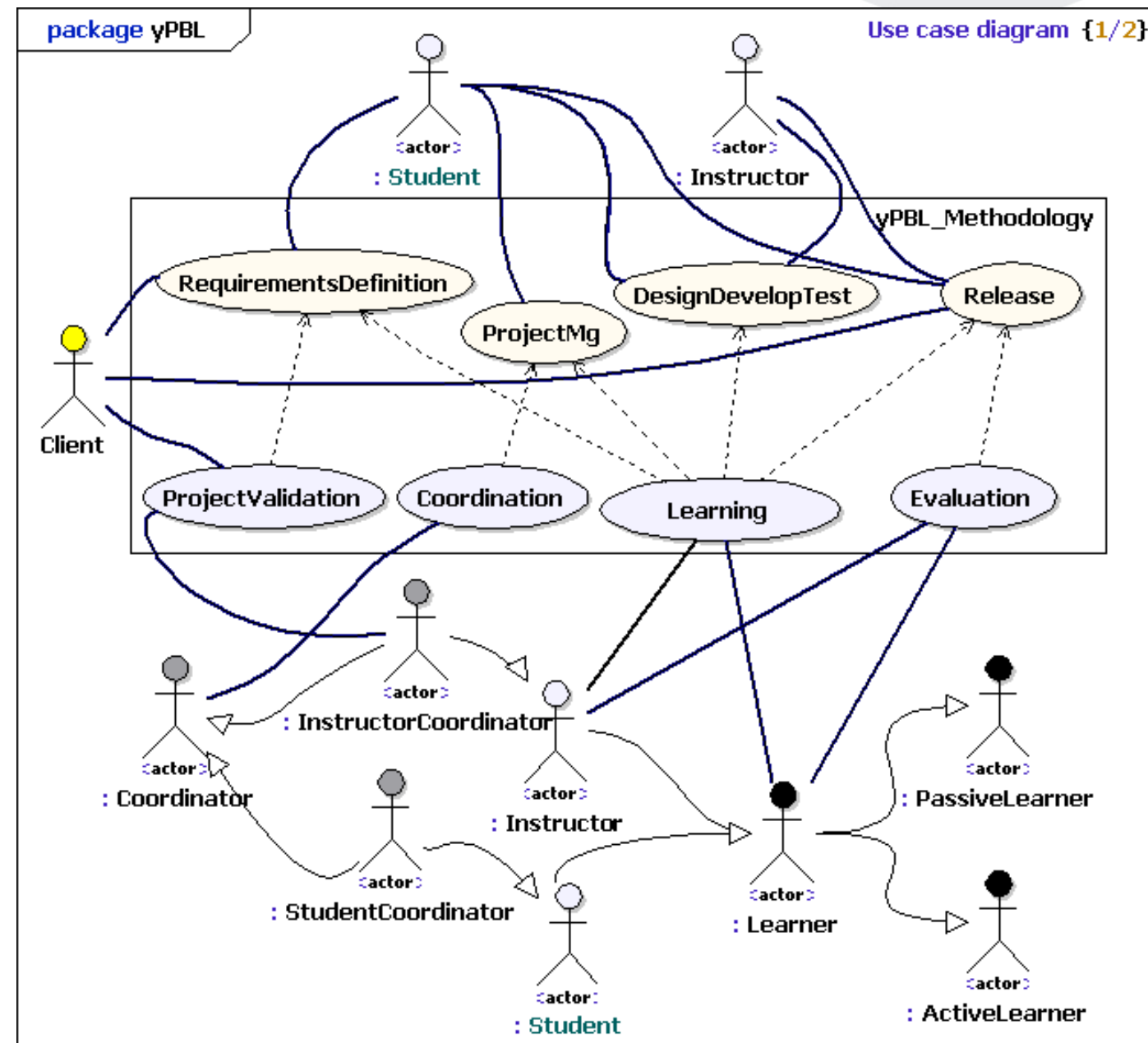
- The **coordinator** role is involved in the **learning project management** and the **learner** in the **learning activities**.
- Specializations of the coordinator role are represented by **instructor coordinator** and **student coordinator** roles. These actors play a supporting role for activities such as **planning, scheduling, hardware and software resources allocation**.



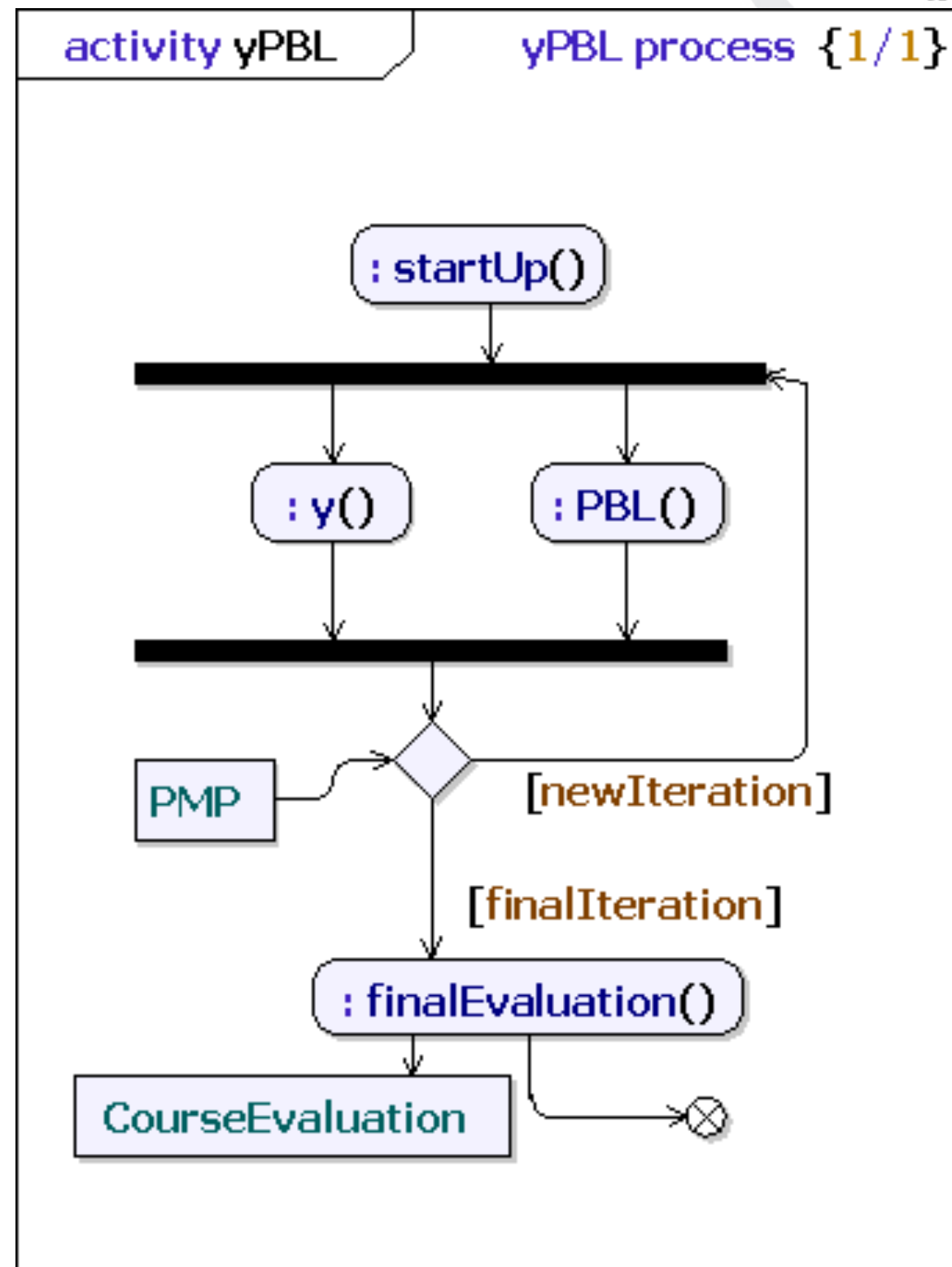
- Moreover, **coordinator** actors **monitor and control the project** in order to early detect potential problems and work together to find efficient solutions. Specifically, the **instructor coordinator** actor is the one interacting with the **external client** in order to **study and validate the project** to be used to instantiate the methodology.



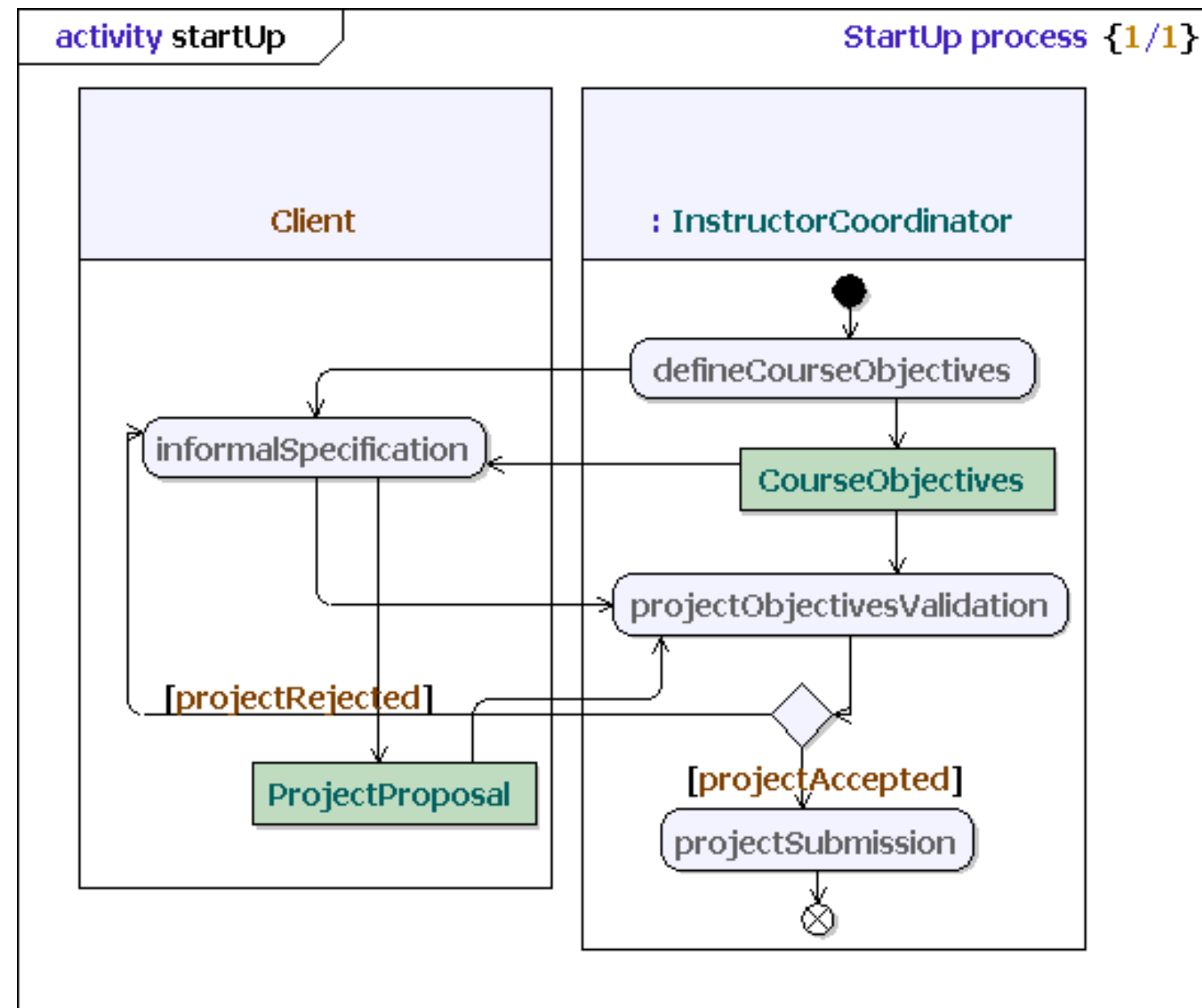
- Generalizations of the **learner** role are defined by **passive** and **active learners** roles. **Students and instructors** play these learning roles.



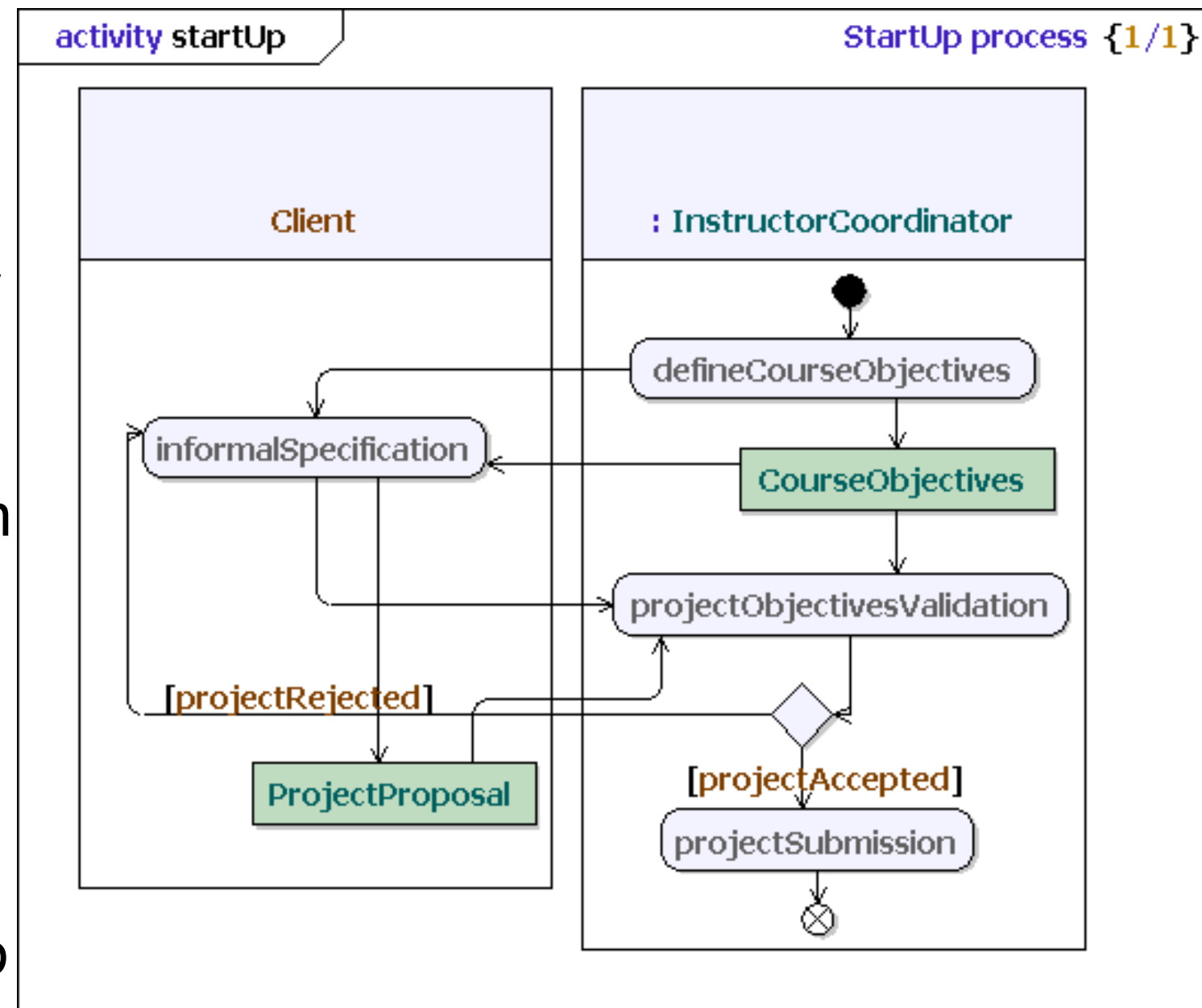
- **start up**: to prepare and validate the overall software-learning project.
- **y process**: concentrates in software engineering activities.
- **PBL process**: targets the learning activities.
- At least 3 iterations.
- **Final evaluation**: learners participants as well as the learning process itself are evaluated.



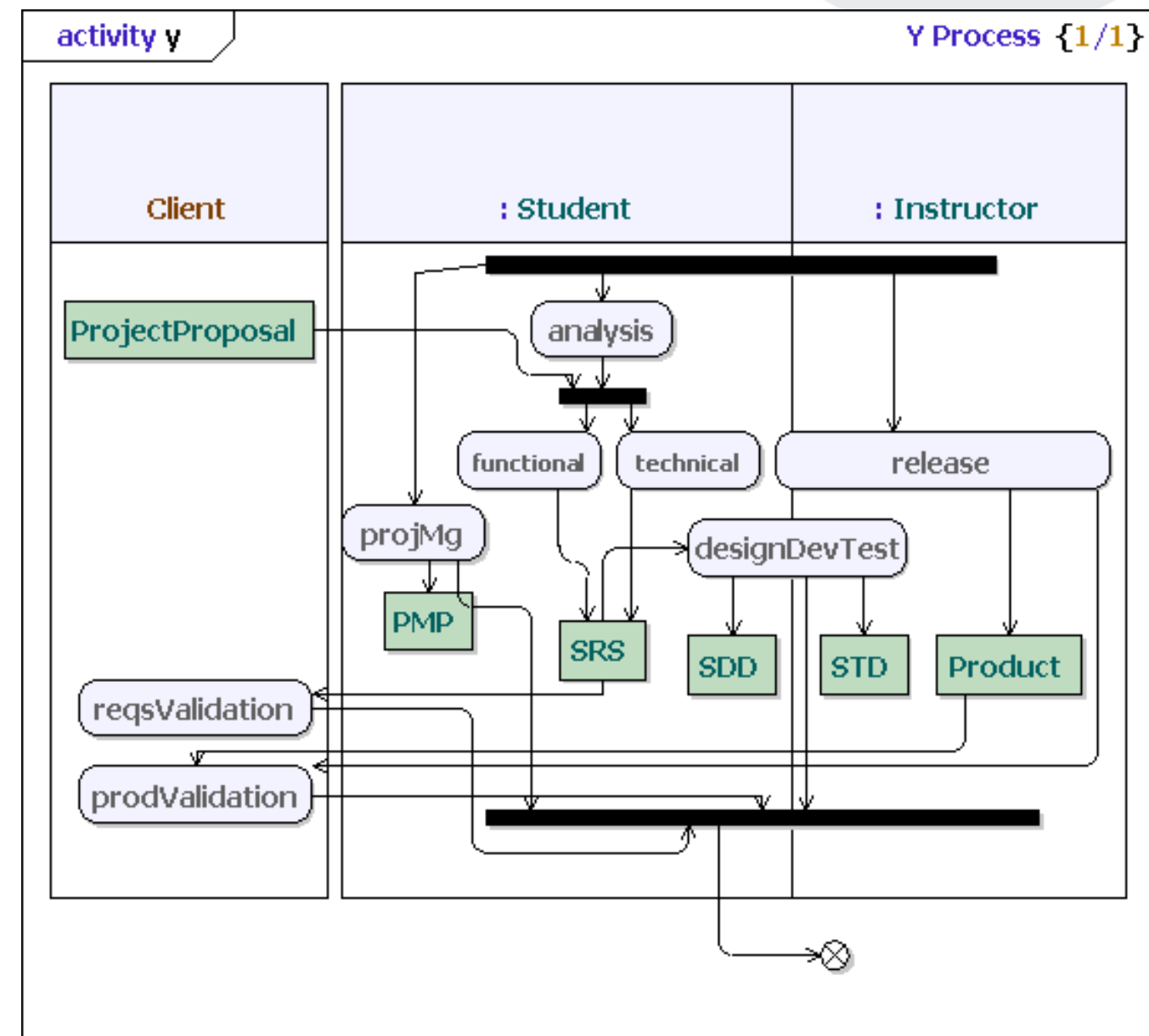
- The instructor-coordinator defines the **course objectives** from the **functional and technical point of view**.
- Functional objectives are defined as abstract learning statements aimed at expressing fundamental **knowledge** goals.
- Technological learning objectives are intended to express concrete statements based on current software technologies to be used by the learners in order to apply the basic knowledge goals (**skills**).



- The document called “**Course Objectives**” is used to collect these functional and technical objective specifications. This document is communicated to potential clients in order to allow them to propose an objective-compliant project.
- **Clients** are asked to propose an informal specification of the project in the form of a “**Project Proposal**” that will be validated or rejected by the instructor-coordinator. If the **project is accepted** it will be **submitted** to the rest of the yPBL process actors.



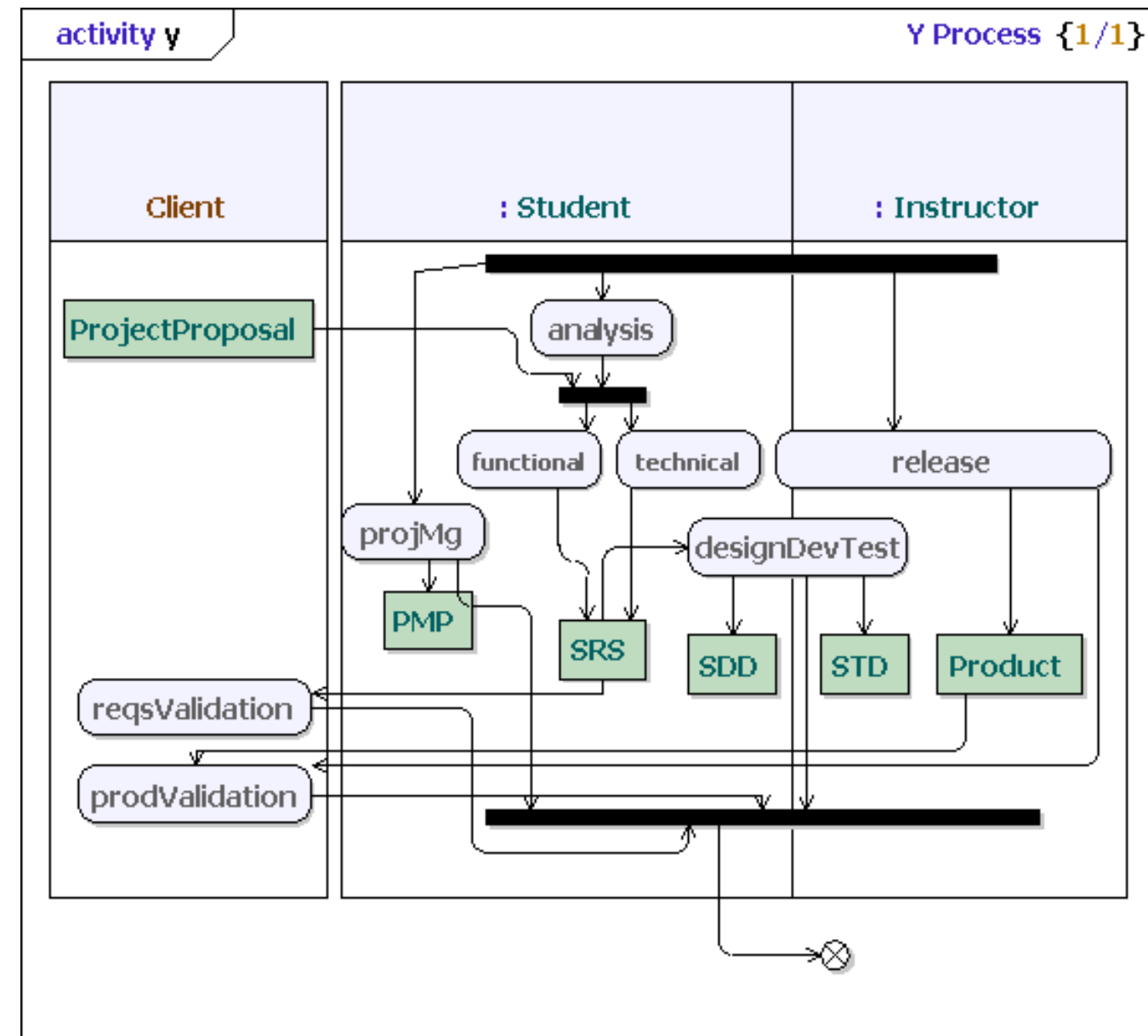
- **software process:** students and instructors perform collaborative or individual activities for every iteration of the Y process.
- In order to stimulate autonomy skills, students are asked to work on the functional and technical **analysis phase** of the project based on the “Project Proposal” submitted by the client.



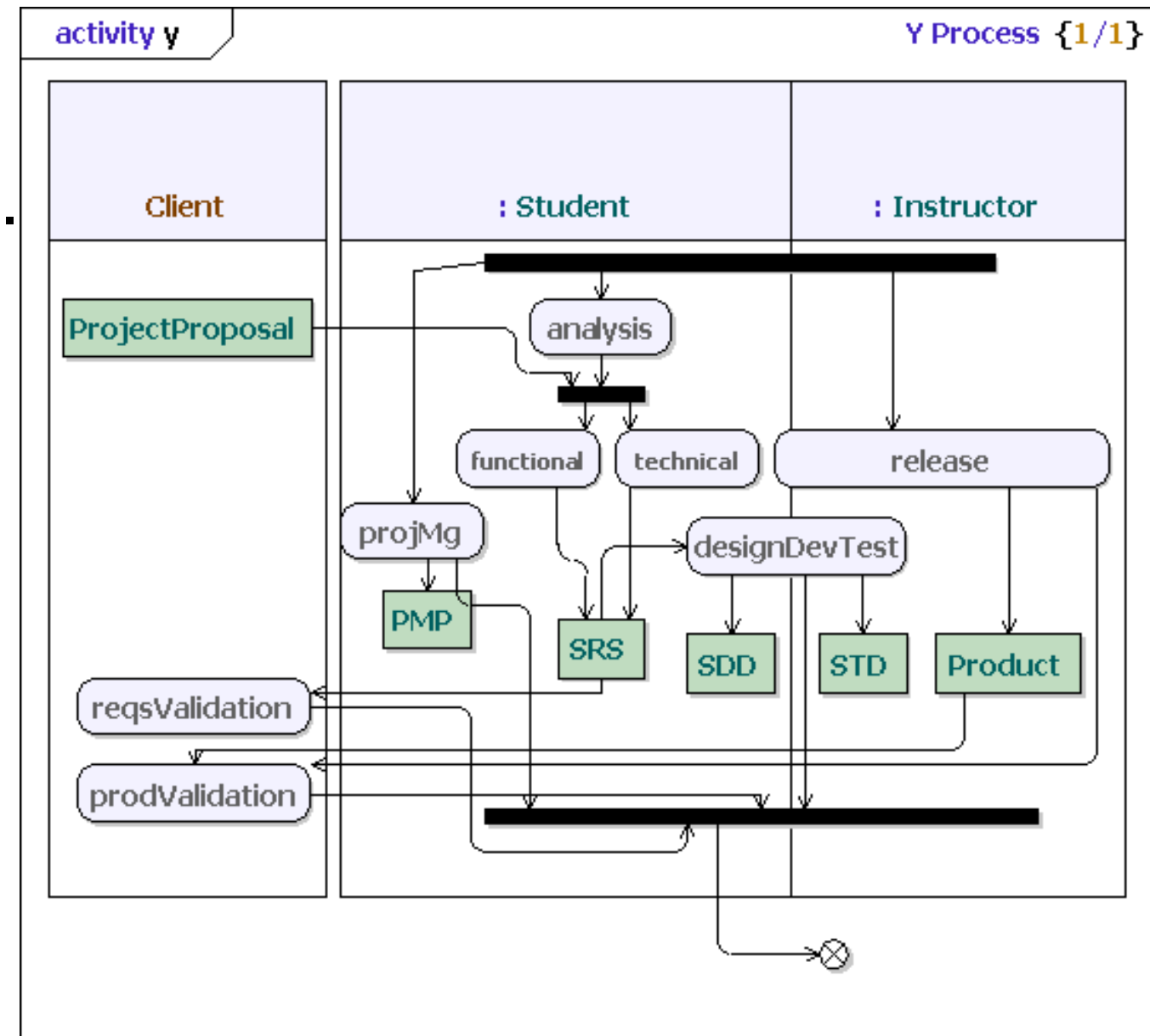
- INSA de Toulouse • 135, Avenue de Rangueil • 31077 Toulouse • France  
Tél. : +33(0)5.61.55.95.13 • Fax : +33(0)5.61.55.95.00 • [www.insa-toulouse.fr](http://www.insa-toulouse.fr)



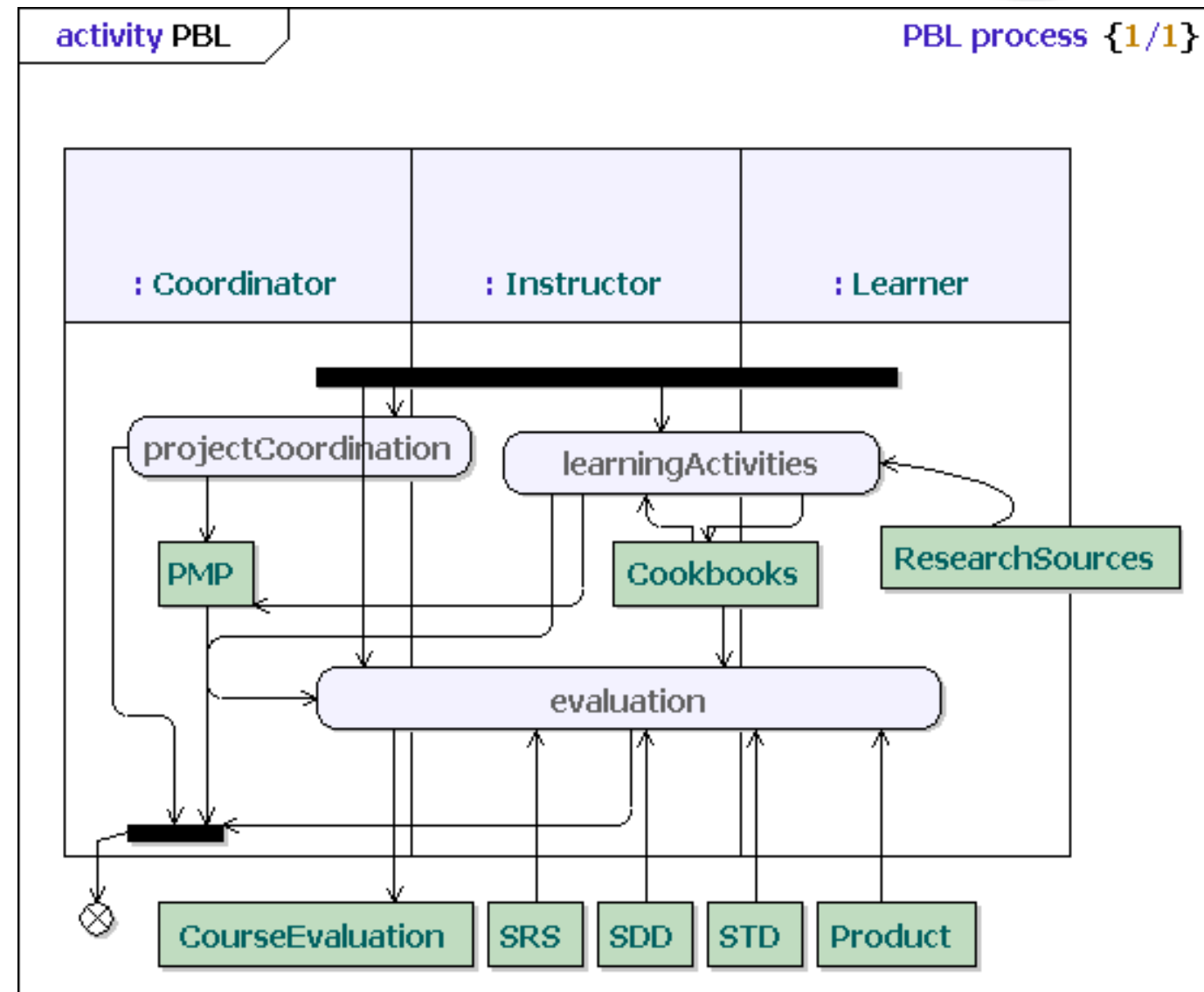
- During the **design, development and testing activities**, both students and instructors work together in order to produce the **artifacts expected to be released in every iteration**. In these activities the role of the instructor is clearly separated from the client role and this is one important benefit offered by the yPBL method. Indeed, **the instructor plays a supporting role** intended to help the students to achieve the software project objectives.



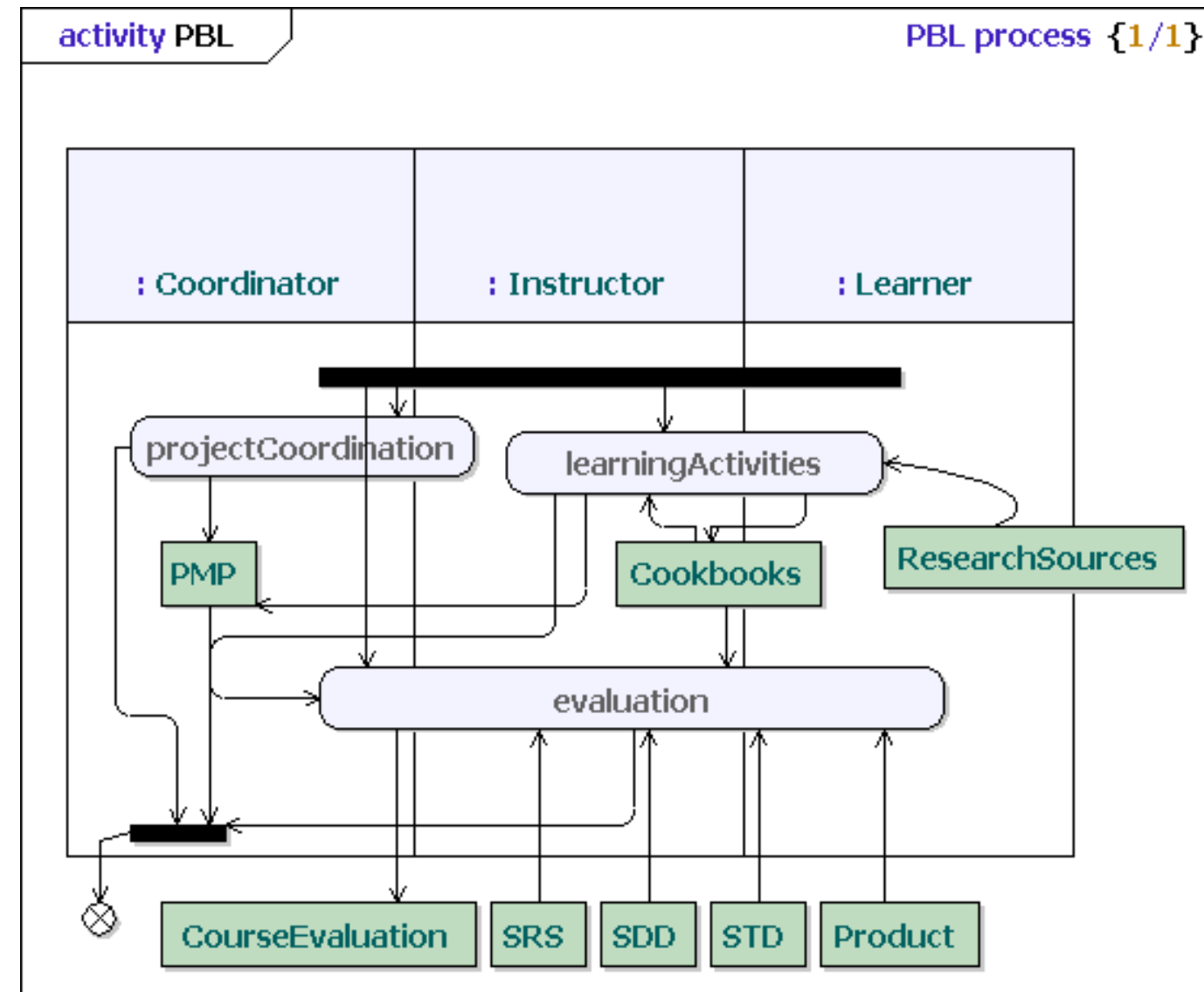
- Design and test oriented documents are produced based on the **SDD** and **STD** standards. Following the **PMP** plan and before the end of the iteration, specific interactions need to be performed with the client in order to **validate** the “**Product**” release against the software requirements expressed in the **SRS**.



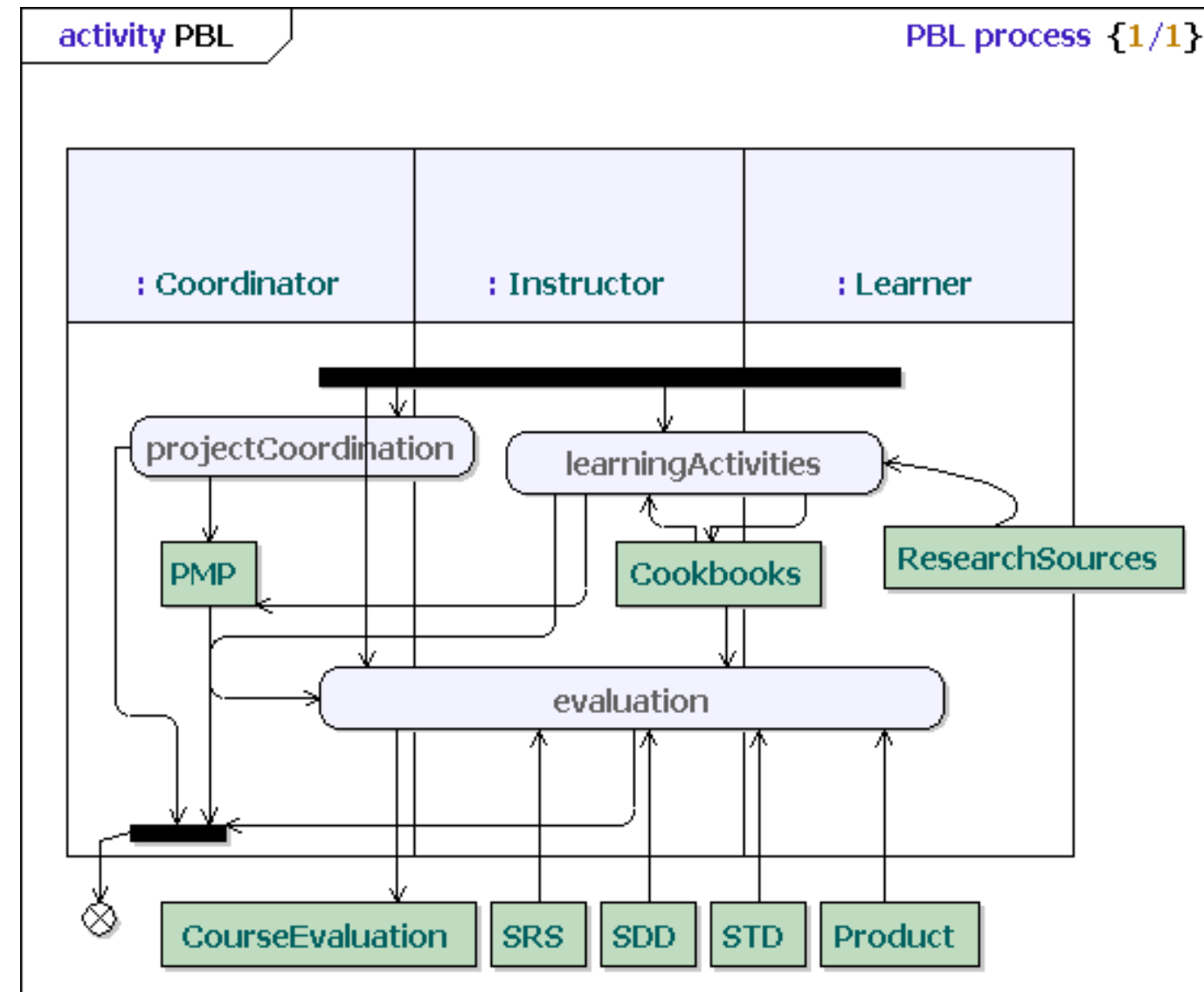
- **Instructors and learners** work together in learning activities, which are naturally deduced from the plan, defined in the PMP: **cookbooks**.
- **Coordinators** work together on the basis of periodic meetings. During these interactions, coordinators exchange monitoring information collected during the process. This information can be used to encourage positive experiences and good practices as well as to work together in solutions to solve specific difficulties.



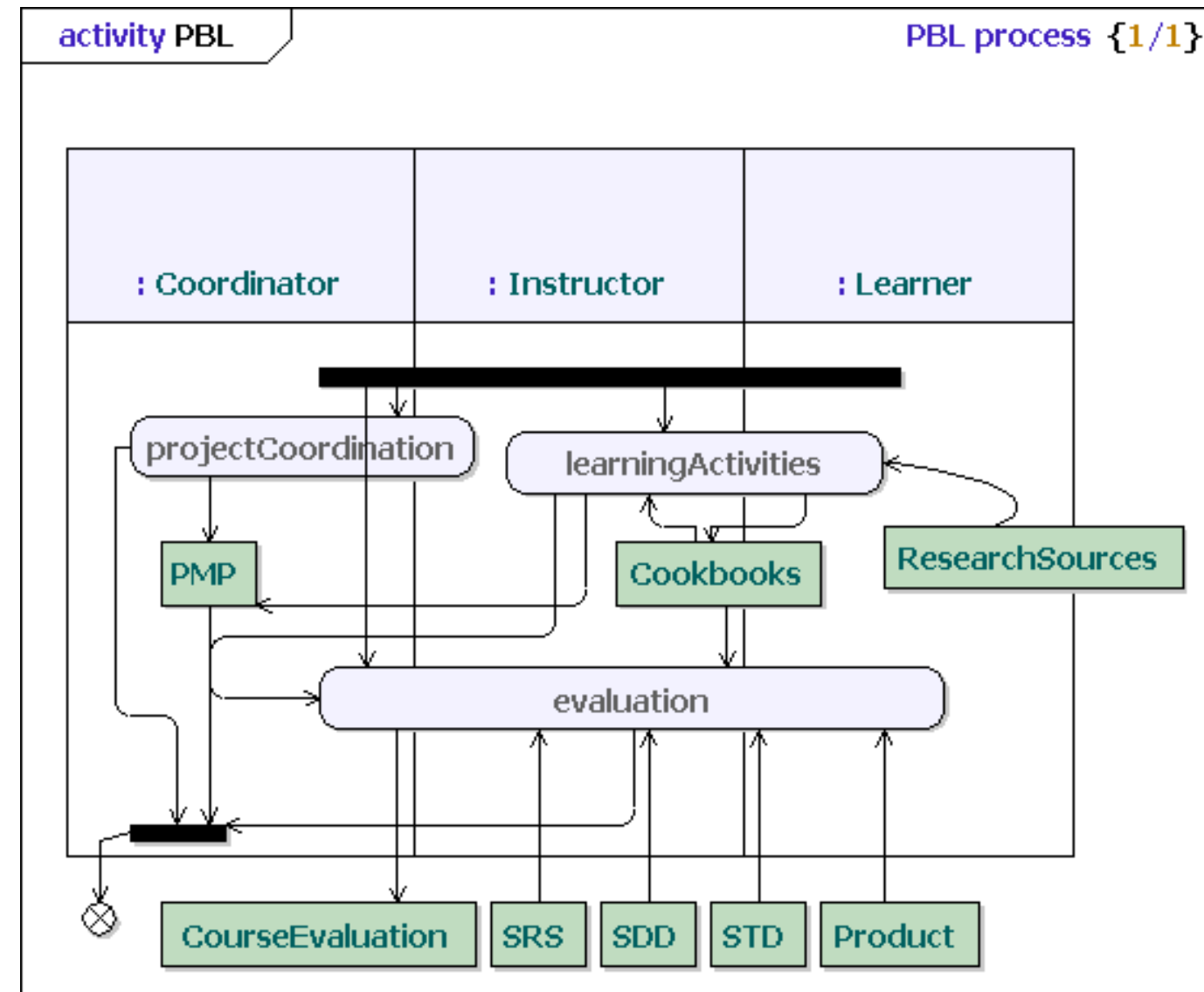
- **Students and instructors** work together to define and plan learning activities to be carried out in every iteration of the process. In order to efficiently carry out these learning activities accordingly with the plan, both actors need to participate in the **research, preparation and review of the learning material**. Instructors work together on the definition of a list of learning subjects. These subjects will be prepared and presented by the students.



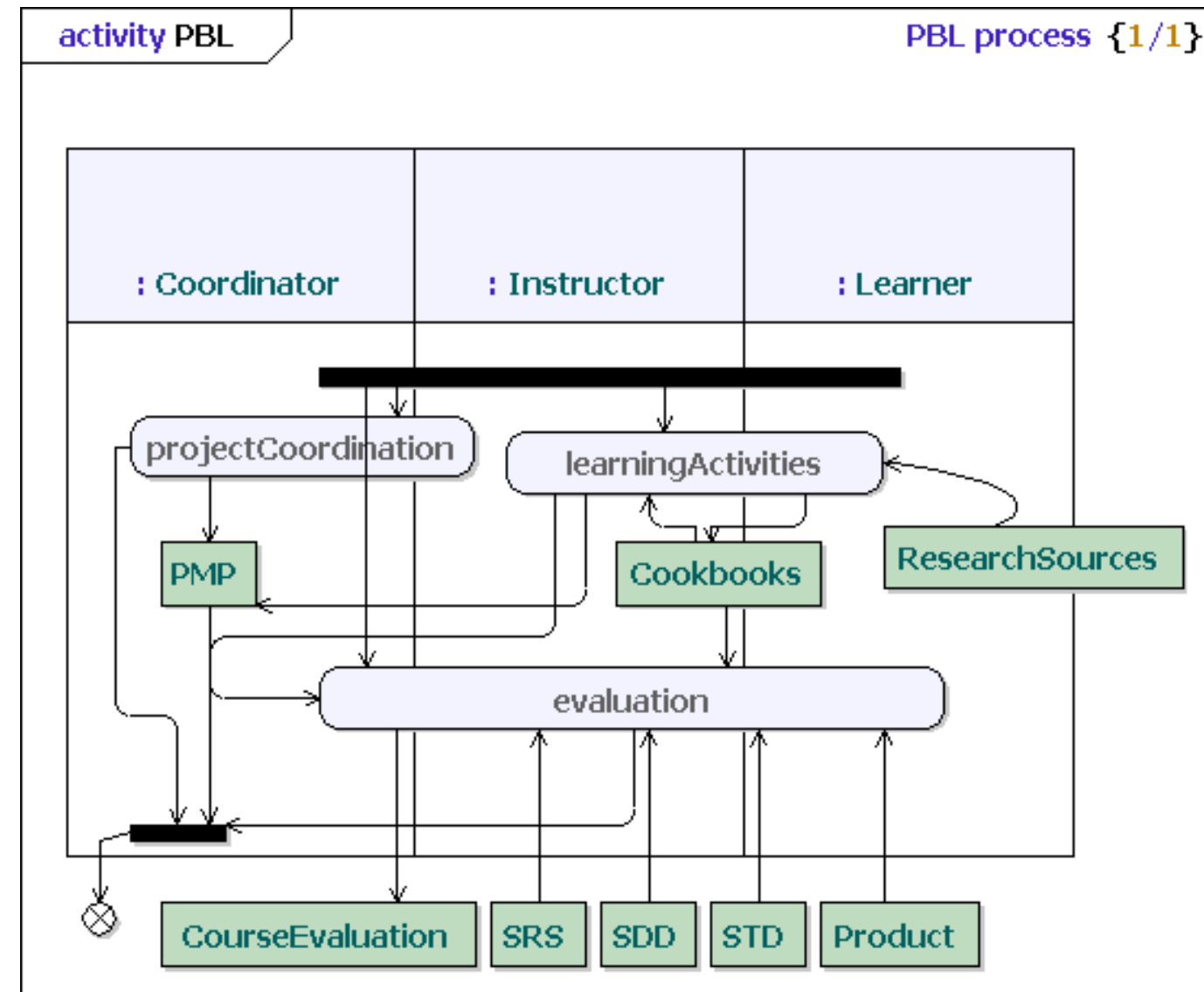
- “**Cookbooks**” are aimed at proposing an efficient presentation of definitions and concepts (i.e. the **ingredients**), and how they can be applied to construct a particular software function or service (i.e. the **recipes**). Recommended resources and links are also proposed in the cookbook.
- Instructors and students carry out the **preparation of the cookbooks** and specific timeslots are reserved to allow them to present these learning materials. The cookbooks are also stored in a **common repository** in order to facilitate its access during the project process.



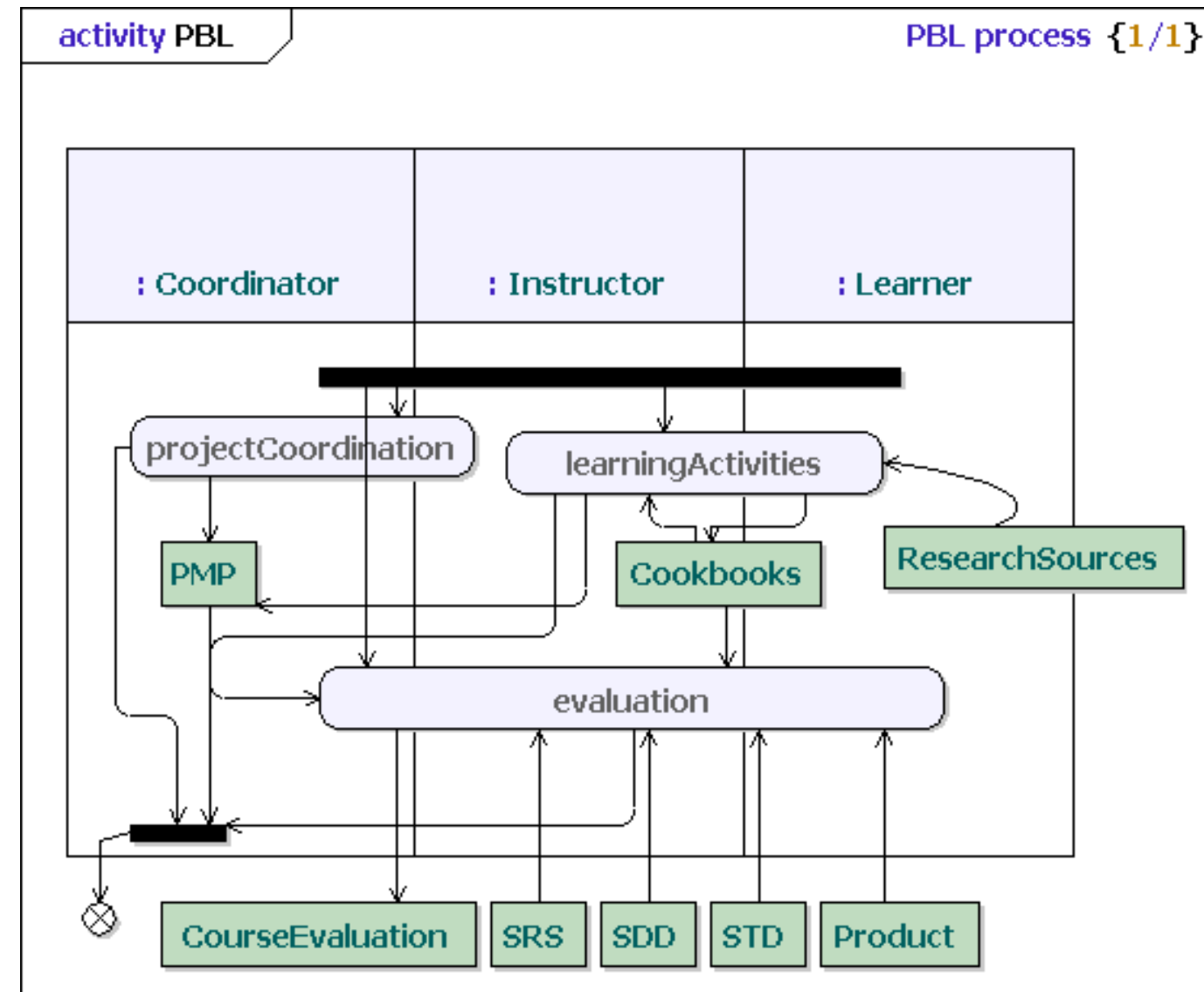
- The evaluation of these activities is carried out by the **peers** based on the **real knowledge acquisition**.
- Cookbooks are presented to all the students and instructors during a **conference** (end of 1st. Iteration).
- Technologies will be selected after this conference and applied to satisfy project requirements during the software building process.
- Skills are developed during 2nd and 3rd iterations (internal actors apply their acquired knowledge in constructing the real software solution).



- Finally, for every PBL iteration the **evaluations** is carried out based on the **SRS, SDD and STD** documents. From these documents the **achievements can be objectively measured** based on the requirements identification, solution design and implementation, and the test performed on the final product. Results of the evaluation in every iteration are stored in the “**Course Evaluation**” document.



- A **final evaluation activity** asking for the participation of all the process actors is performed after the final **project presentation**. During this activity, the **functional and technical project requirements are finally measured**, as well as the global satisfaction of the internal and external actors. **The process itself is discussed and a list of suggestions and remarks are compiled in the “Course Evaluation” document.** This information is very helpful to improve the process for future projects.



### Active

- apprentissage des connaissances scientifiques/techniques (Cookbooks) et compétences (Recettes)
- communication – interaction avec le client, à l'intérieur et entre les groupes, utilisation de ressources écrites, rédaction et présentation des cookbooks, appropriation et explication de sa partie à des experts et non-experts
- prise de responsabilité au sein du groupe (coordinateur du projet, architecte logiciel, développeur, testeur, chef de qualité...)

### Collaborative

- Cookbooks soumis à révision par les paires et les enseignants avant d'être publiés
- "Cookbook Conference"
- réalisation d'un produit final en groupe de 24 divisé en sous-équipes avec différentes responsabilités

### Multi-disciplinaire

- ingénierie logicielle, technologies logicielles, anglais

### Apprentissage durable

- renforcement des connaissances et compétences à travers leur réapplication dans un véritable projet d'ingénierie logicielle

### Préparation pour la vie professionnelle : « working in the clouds »

- utilisation de l'anglais comme langue de travail (avec des non-anglophones), des plateformes collaboratifs (type GoogleDocs), des documents standardisés dans l'industrie (IEEE)

## 9 yPBL projects since 2009 (students of 4th and 5th years of "Informatique et Réseaux" engineering)

### Exemple 2011

Subject : a service-oriented web application for scholarship management for the INSA DGEI department.

Actors :      72 students (6 groups de 12)  
                 4 SE teachers  
                 7 SOA-JAVA teachers  
                 3 English teachers

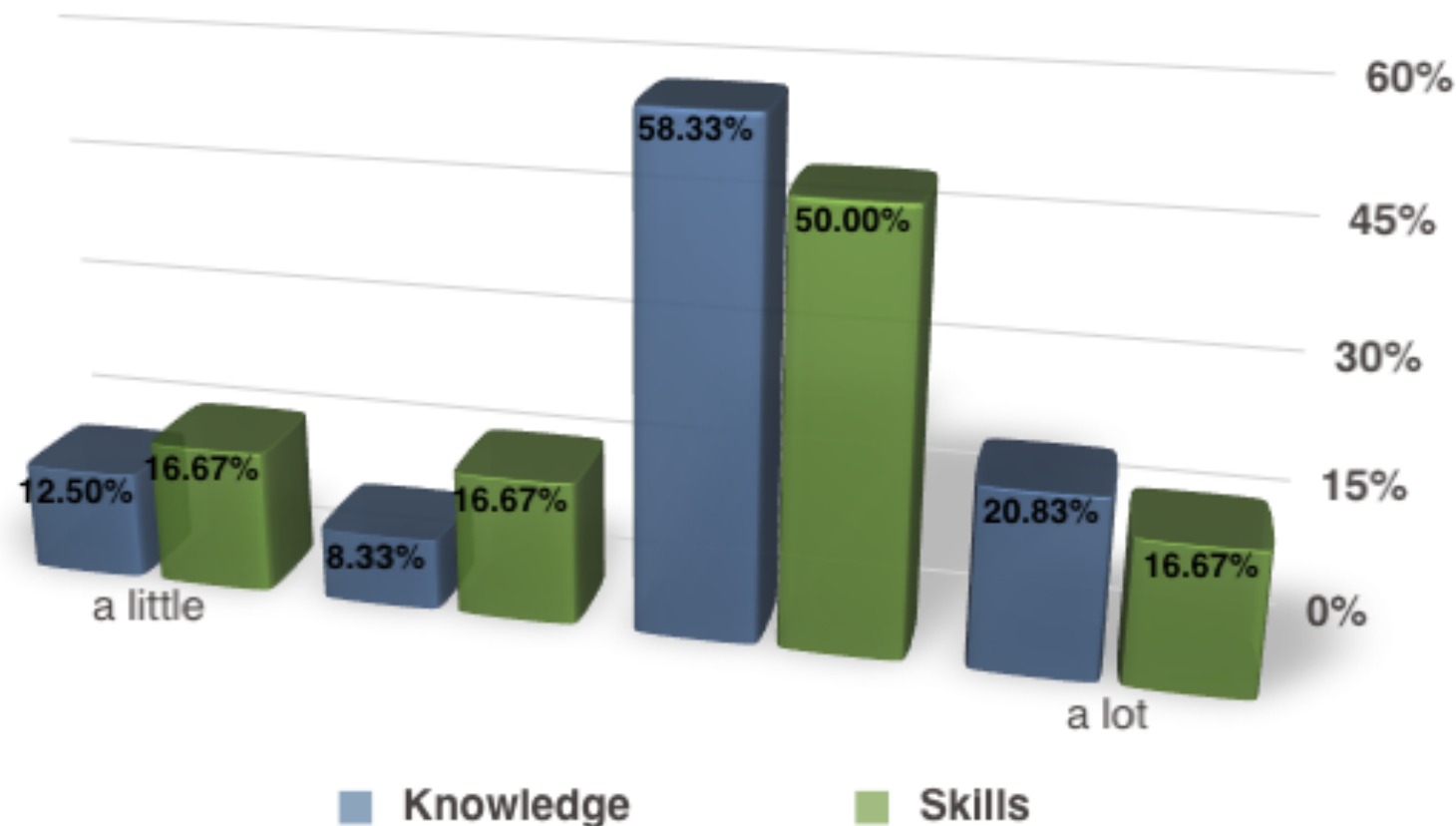
Hours : 115

Iterations : 3

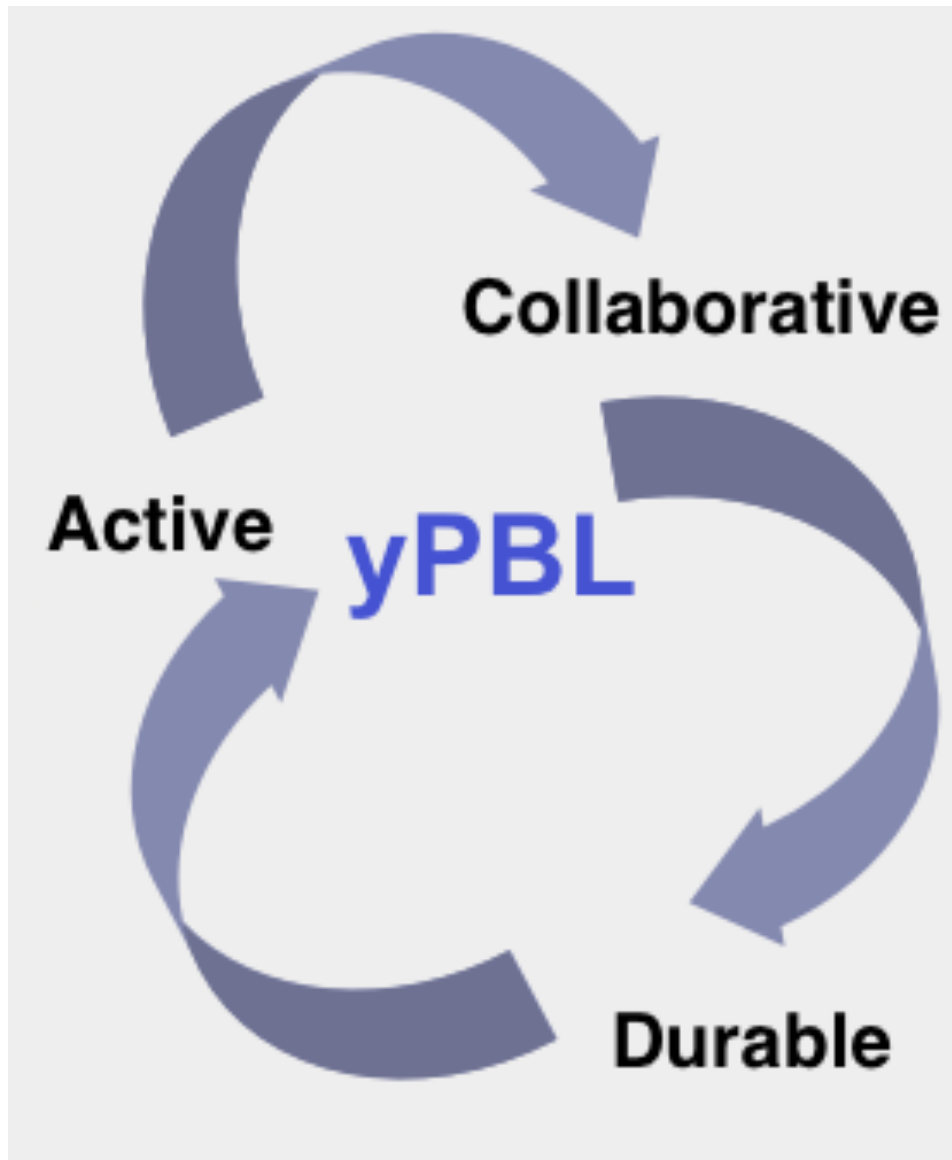
Cookbooks : 32 (3-4 recipes/cookbook)

ECTS :15

## Knowledge and skills acquired by the students



Perception of students (january 2011)



## yPBL = professional process + learning methodology

- Better acquisition of knowledge and skills
- Motivation improved of students and teachers
- High autonomy
- Self-Assessment based in real use cases

**The development of a real complex project leads to face to difficulties and frustrations: a valuable experience and confidence reinforcement.**

**Active, collaborative and durable learning process**