The 2nd International Workshop on Education through
Advanced Software Engineering and Artificial Intelligence

EASEAI
2020

Agile and Cyclic Learning in Teaching Parallel and Distributed Computing

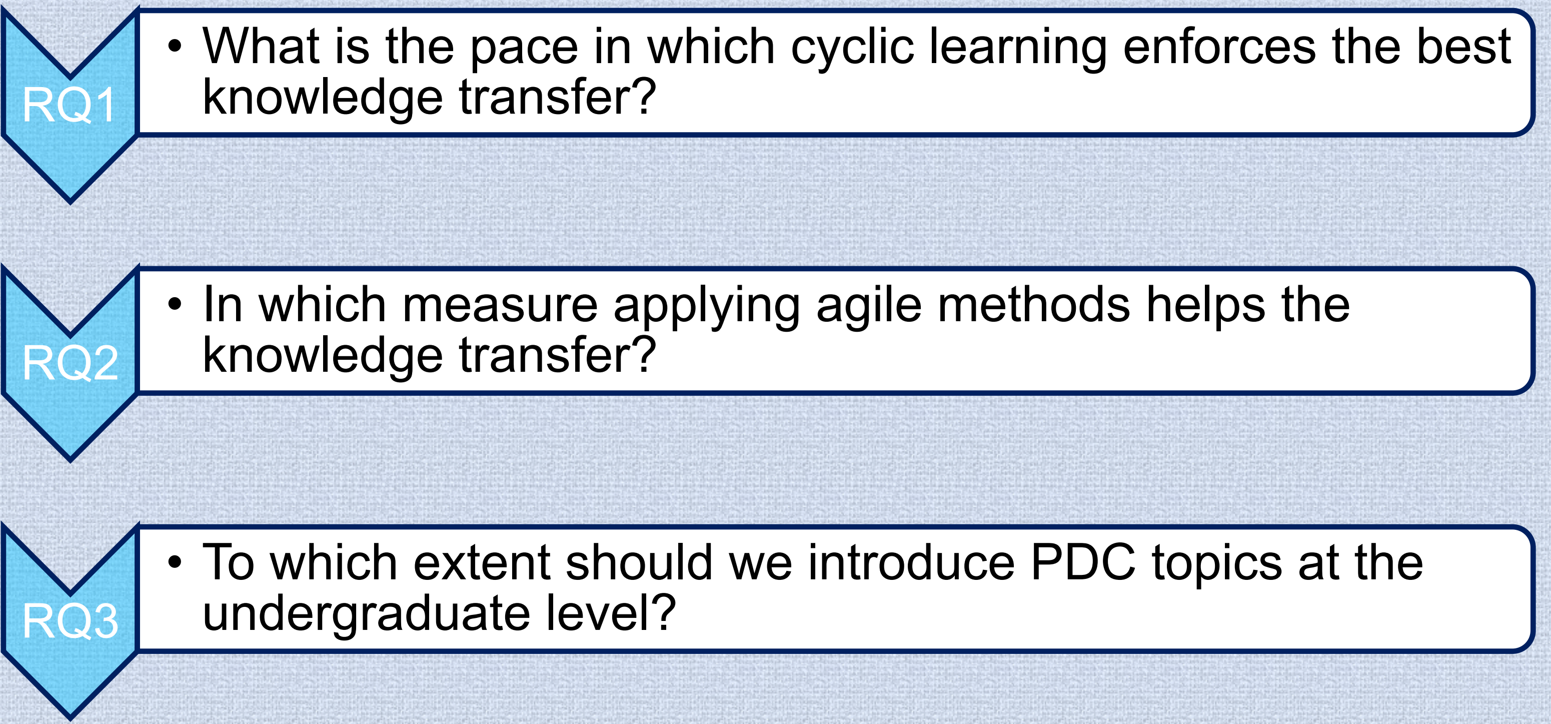
Virginia Niculescu & Adrian Sterca & Darius Bufnea

Babeș-Bolyai University, Romania

Motivation and Goal

- To improve the teaching process of Parallel and Distributed Computing
- To use Agile and Cyclic Learning
- To evaluate the pace in which Agile and Cyclic Learning enforces the best knowledge transfer of PDC concepts

Research Questions



RQ1

- What is the pace in which cyclic learning enforces the best knowledge transfer?

RQ2

- In which measure applying agile methods helps the knowledge transfer?

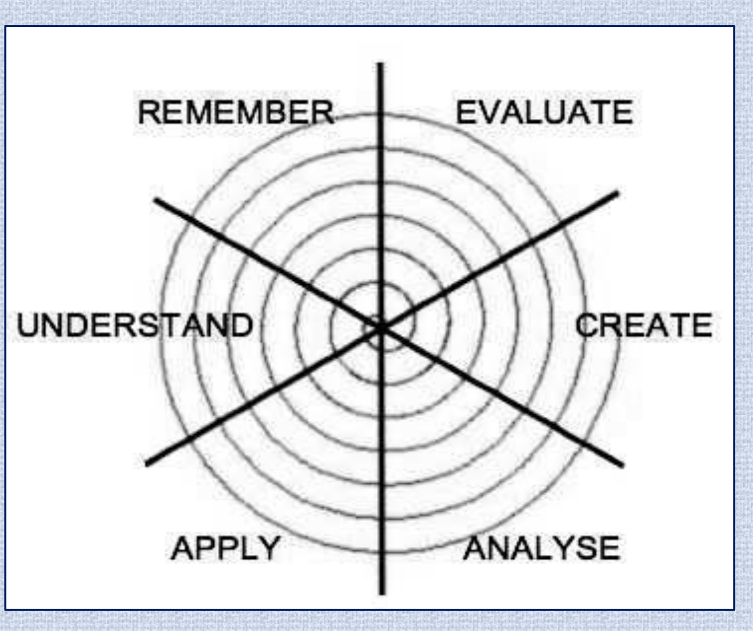
RQ3

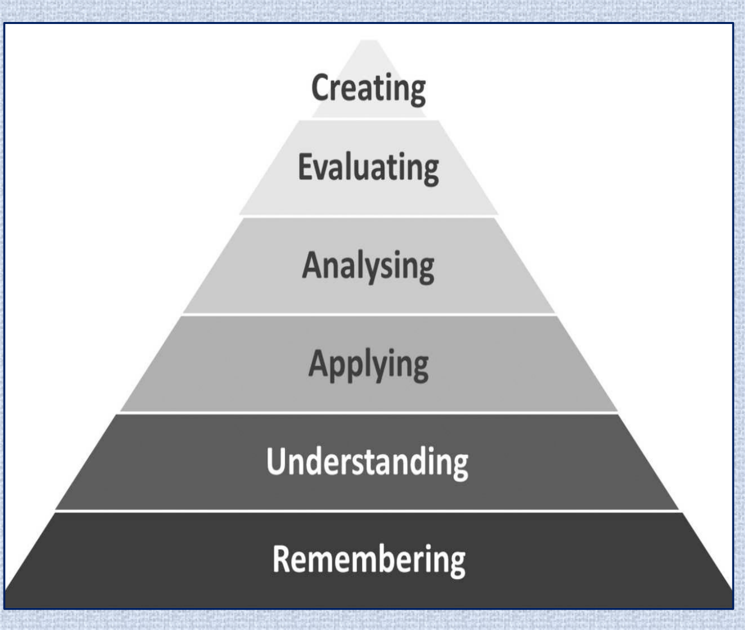
- To which extent should we introduce PDC topics at the undergraduate level?

Learning Methodologies

Cyclic Learning

Bloom taxonomy



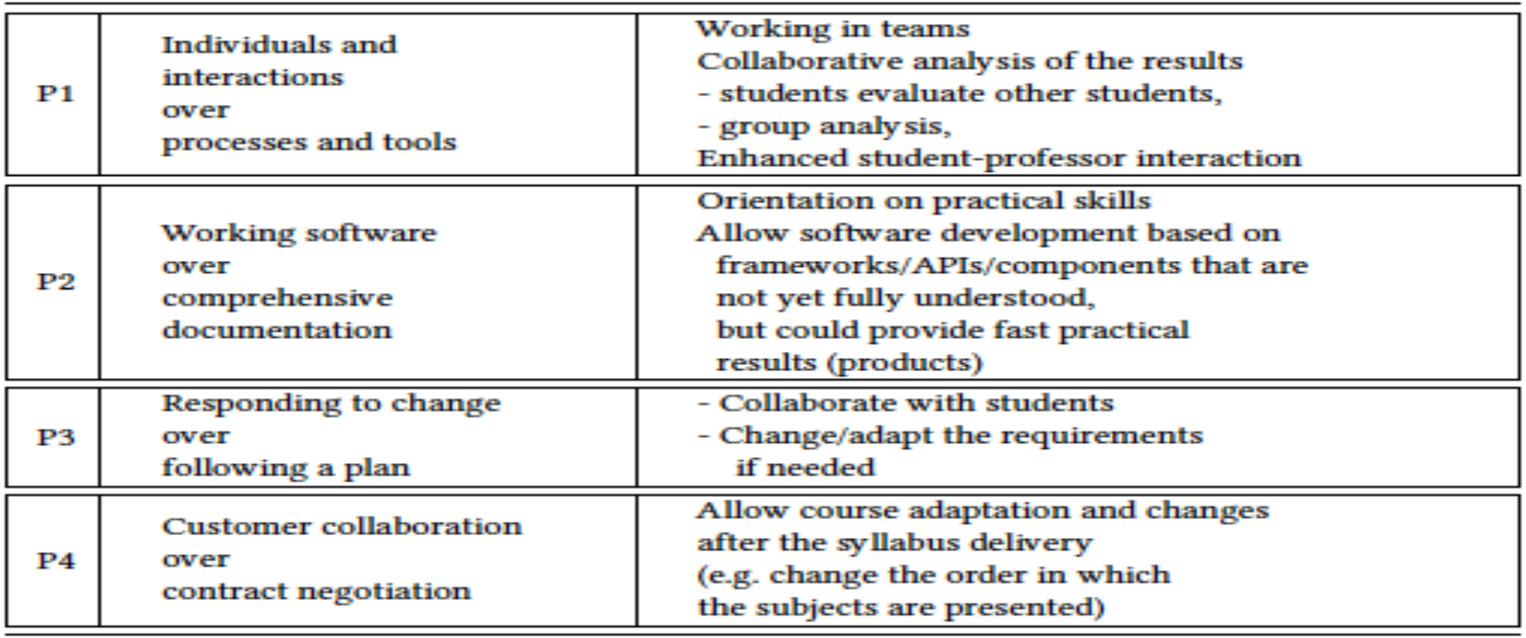


Learning methodologies

Agile Learning

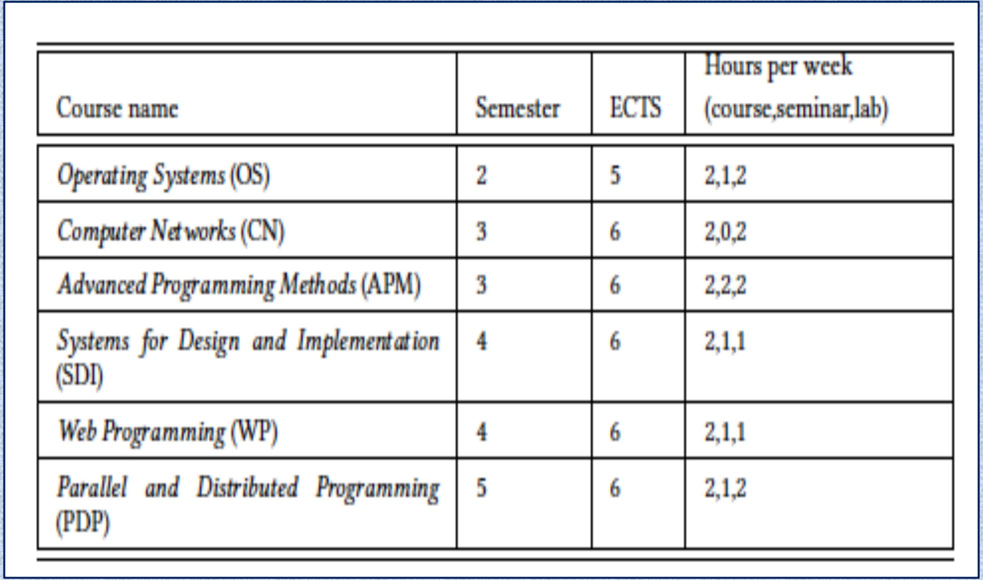
Agile Principles | **Learning Principles**

Agile Manifesto



P1	Individuals and interactions over processes and tools	Working in teams Collaborative analysis of the results - students evaluate other students, - group analy sis, Enhanced student-professor interaction
P2	Working software over comprehensive documentation	Orientation on practical skills Allow software development based on frameworks/APIs/components that are not yet fully understood, but could provide fast practical results (products)
P3	Responding to change over following a plan	- Collaborate with students - Change/adapt the requirements if needed
P4	Customer collaboration over contract negotiation	Allow course adaptation and changes after the sy llabus delivery (e.g. change the order in which the subjects are presented)

Courses addressing PDC topics
cyclic learning



Course name	Semester	ECTS	Hours per week (course,seminar,lab)
Operating Systems (OS)	2	5	2,1,2
Computer Networks (CN)	3	6	2,0,2
Advanced Programming Methods (APM)	3	6	2,2,2
Systems for Design and Implementation (SDI)	4	6	2,1,1
Web Programming (WP)	4	6	2,1,1
Parallel and Distributed Programming (PDP)	5	6	2,1,2

- ✓ one item is introduced in successive iterations
- ✓ more than one iteration is needed in order to attain the highest level of knowledge
- ✓ students return to previously learned concepts:
  - with regularity
  - in different contexts

  each time they extend and deepen their knowledge

Agile Learning use-cases

APM -> P2 principle

- working with executors, tasks, and futures without explaining before the associated concepts
- use Java parallel streams without discussing about the mechanisms through which the parallelism is achieved in this case

WP -> P1 and P3 principles

- students team up in group of 3 or 4 in order to deliver their assignments.
- after each sprint -> double evaluation:
  - an individual evaluation of student's knowledge
  - a group assessment

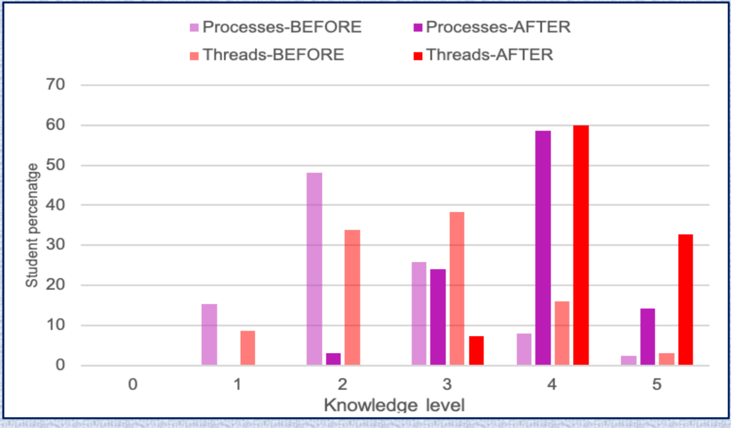
PDP -> P1, P3 and P4 principles

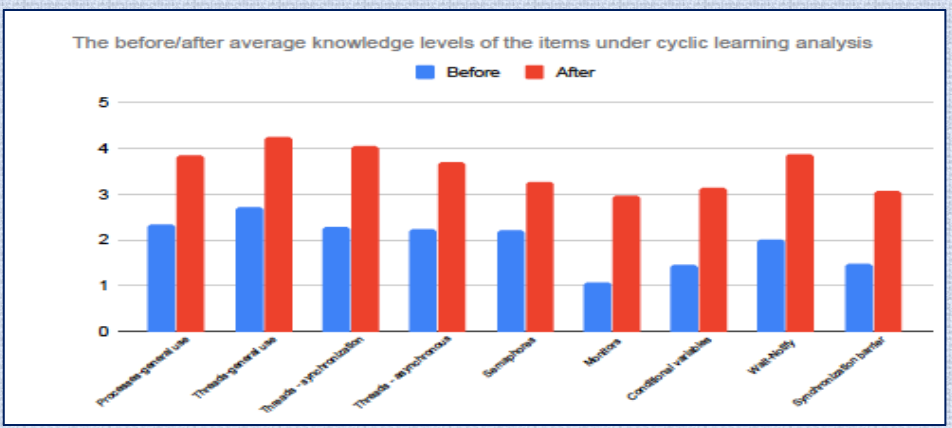
- (**P4**) MPI (Message Passing Interface) presentation re-scheduling based on the students' feedback (to increase interest)
- **(P1)** CUDA programming practical work was set to be done in teams (to overcome the difficulties)
- **(P3)** A project-based learning alternative evaluation (challenge problems)

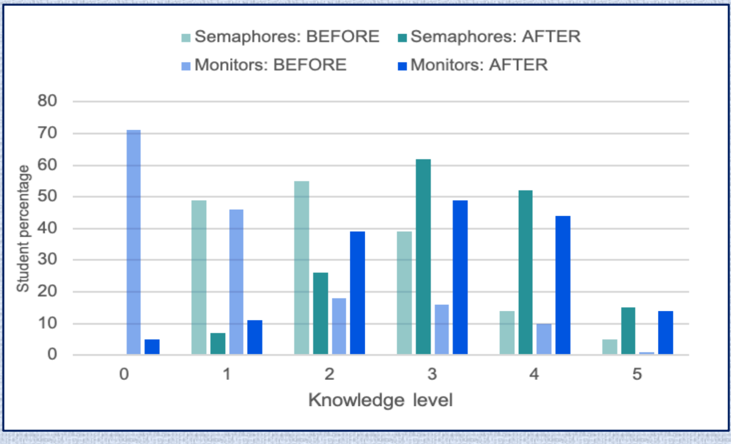
Analysis

- Questionnaire Based Investigation
  - student self evaluation – 6 levels (0- Not known ⇔ 5 –assessment)
  - student informal feedback
- Grade Analysis
  - Laboratory works
  - Theoretical written exam
- where/when
  - the end of the PDP course which is the last in the intercorrelated courses chain

Questionnaires



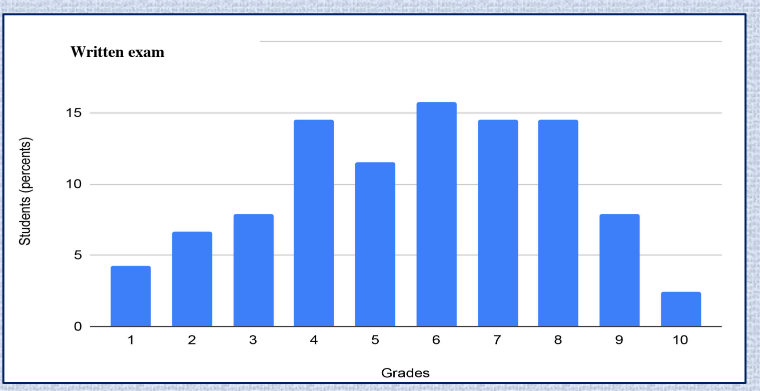


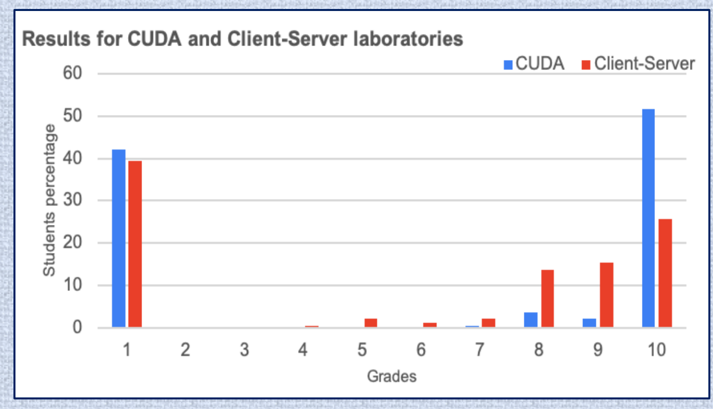


Informal qualitative feedback:

- Team work very much appreciated! (web progr., and others...)!
- CUDA enjoys a large interest from students, but also MPI and OpenMP.
- MPI considered helpful for deepening understanding of processes.
- Most of the students expressed the fact that the PDC topics are difficult to understand.
- Few others expressed their opinion that there is unnecessary repetition of some items.

Grade analysis





Fundamental knowledge

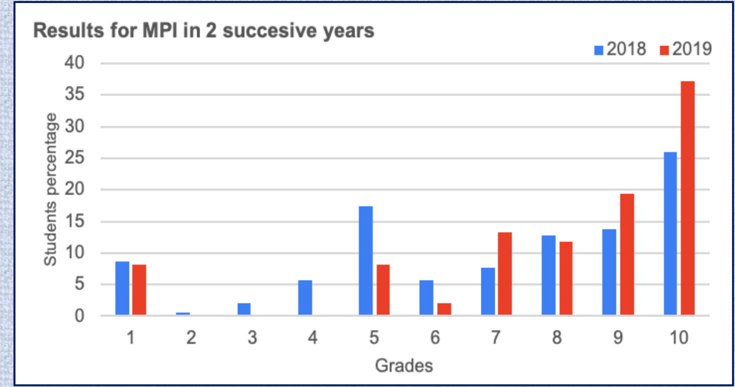
- written exam

Agile learning:

- CUDA– working in teams
- MPI course – presentation order changed

Cyclic learning:

- Client-Server – still considered difficult



Conclusions

Response to RQ1 – Cyclic learning

- proved to be an effective and efficient method to be applied for teaching
- the concepts and mechanisms of PDC are not very easy to be completely understood and assimilated (one course wouldn’t be enough).

Response to RQ2 - Agile learning

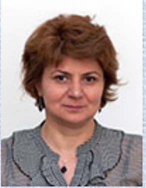
- improve the knowledge transfer
- increase of the students’ interest

Response to RQ3 – Level of knowledge

- we cannot increase very much the pace of introducing PDC topics at the undergraduate level since we have to assure the fact that the fundamentals are well understood and assimilated
- the elective courses could enlarge this knowledge

Thank you for your attention!

Niculescu Virginia:
vniculescu@cs.ubbcluj.ro
virginia.niculescu@ubbcluj.ro



Darius Bufnea:
bufny@cs.ubbcluj.ro
darius.bufnea@ubbcluj.ro

Adrian Sterca:
forest@cs.ubbcluj.ro
Adrian.sterca@ubbcluj.ro

?

Questions