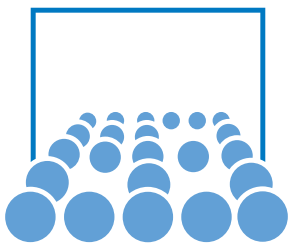
Teamwork Across Disciplines: High-Performance Computing Meets Engineering

Euro-EDUPAR

Philipp Neumann, Christoph Kowitz,
Felix Schranner, Dmitrii Azarnykh

24.08.2015



Outline

Interdisciplinary Education and Teamwork

Course Curriculum

Teamwork Across Disciplines: Concept
Example: Turbulent Flow Simulation on HPC-Systems
Evaluation

Summary & Conclusion

SPPEXA: 1648 Software for Exascale Computing

- Priority programme of the German Research Foundation (DFG)
- Currently 13 projects, each consisting of 2-5 research consortia
- Fundamental research on various HPC software aspects to get prepared for exascale computing
- SPPEXA-2: 2016 – 2018 in cooperation with France (ANR) and Japan (JST)
- Further activities
  - Project-spec. workshops
  - Annual assemblies
  - Educational activities
  - Gender activities and childcare funding

Interdisciplinary Education and Teamwork (1)

Why do we need it: "Computer science/HPC + X"

→ computational science and engineering, bioinformatics, biomedical computing, big data, social media,...

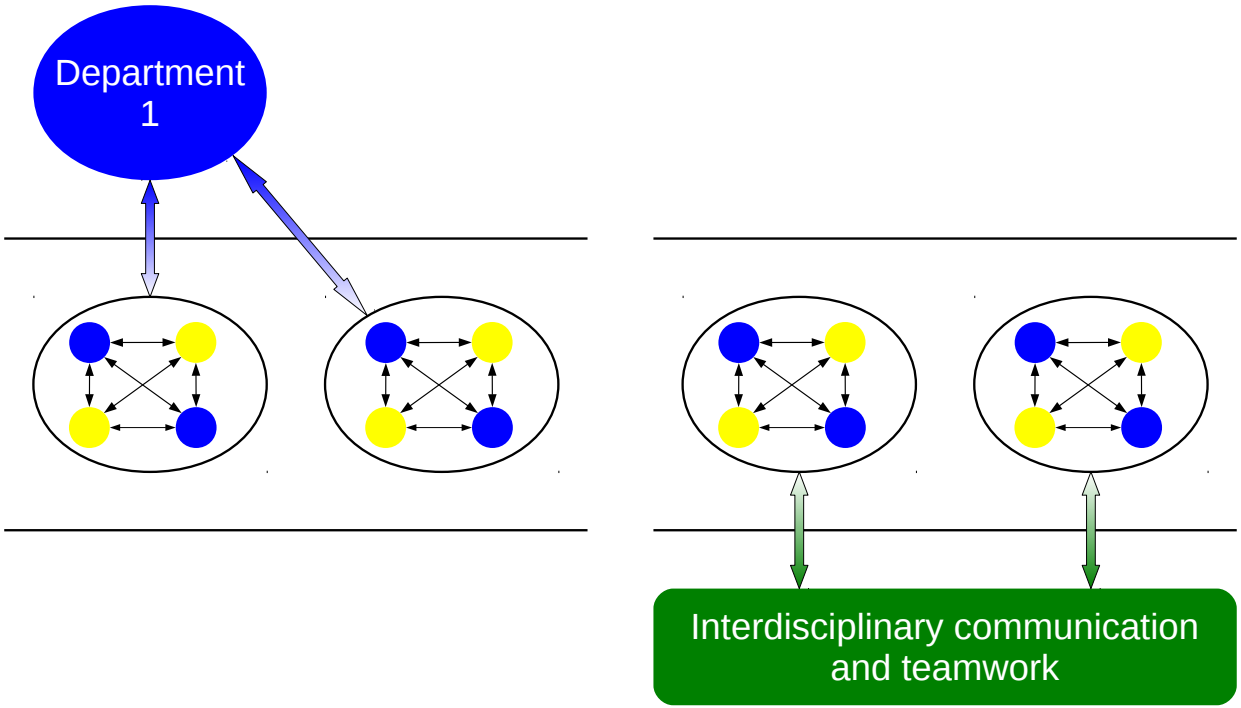
Interdisciplinary Education and Teamwork (1)

Why do we need it: "Computer science/HPC + X"
→ computational science and engineering, bioinformatics, biomedical computing, big data, social media,...

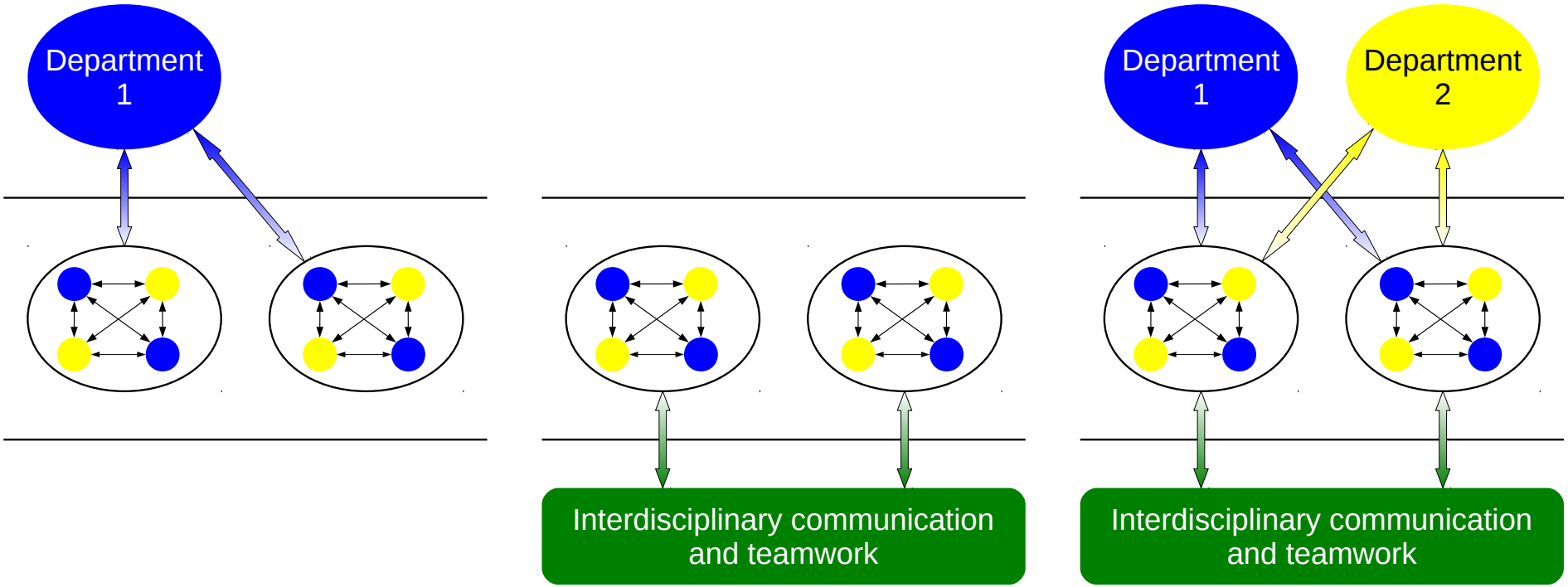
What is "interdisciplinary"?
Challenges (in teaching):

- **C1** "Really interdisciplinary" topic
  → invoke knowledge transfer
- **C2** Involve both disciplines to similar amount
  → stimulate bi-directional knowledge transfer + learning + discussions
- **C3** Prerequisites: Basic understanding of both disciplines + soft skills

Interdisciplinary Education and Teamwork (2)



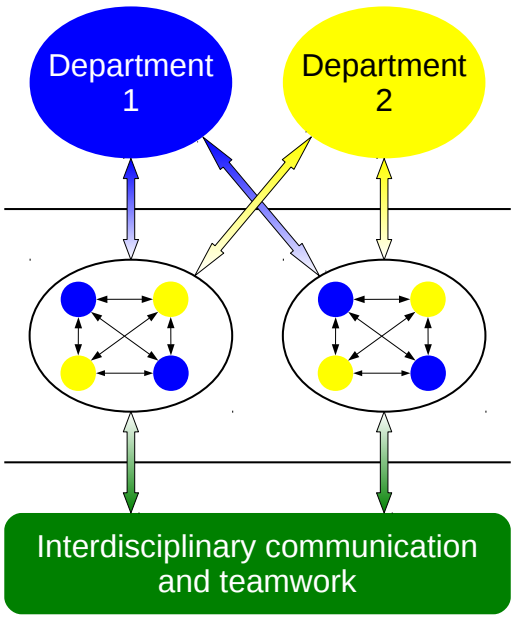
Interdisciplinary Education and Teamwork (2)



Teamwork Across Disciplines

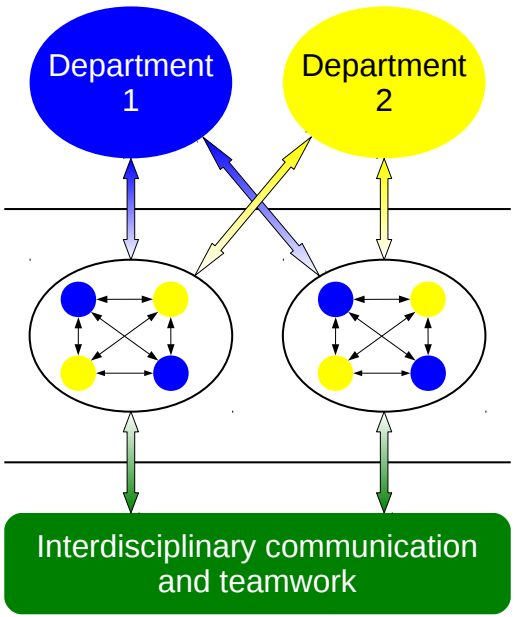
Teamwork Across Disciplines: Concept

- Restriction to two disciplines *D*1, *D*2

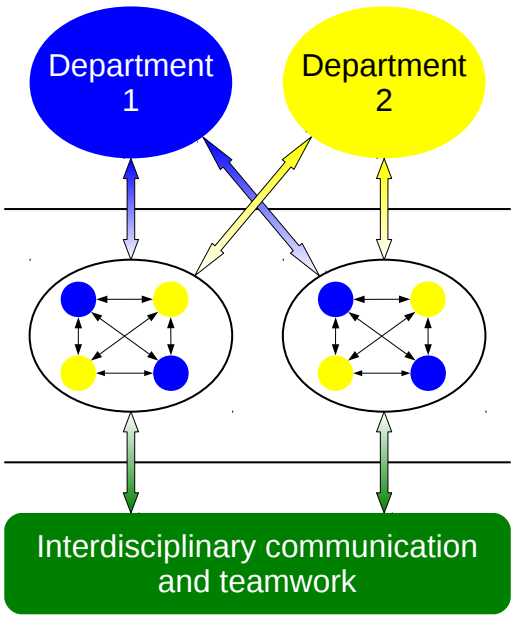


Teamwork Across Disciplines: Concept

- Restriction to two disciplines $D1$, $D2$
- Design interdisciplinary problem P of (overlapping) sub-problems P_{D1}, P_{D2}

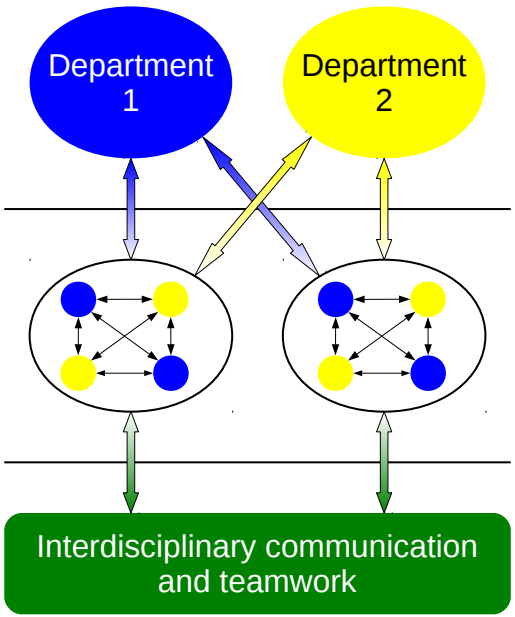


Teamwork Across Disciplines: Concept



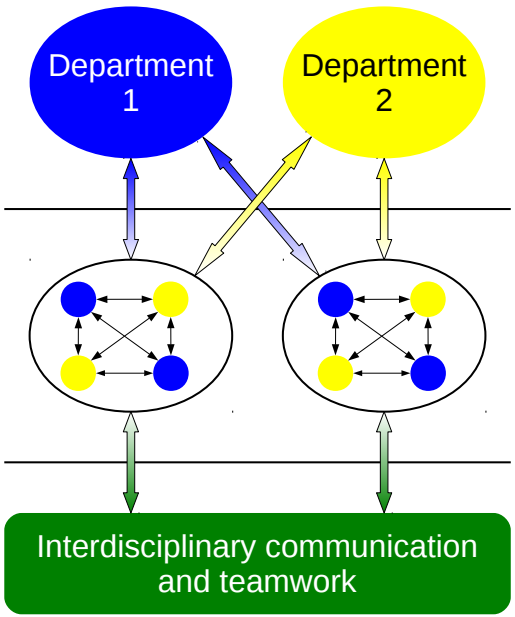
- Restriction to two disciplines $D1$, $D2$
- Design interdisciplinary problem P of (overlapping) sub-problems P_{D1}, P_{D2}
- Form mixed $D1$/$D2$ student teams (2-4 students) to solve P

Teamwork Across Disciplines: Concept



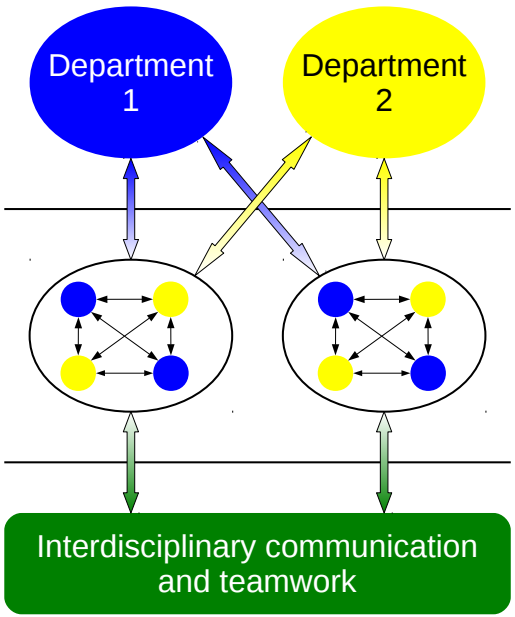
- Restriction to two disciplines $D1$, $D2$
- Design interdisciplinary problem P of (overlapping) sub-problems P_{D1}, P_{D2}
- Form mixed $D1$/$D2$ student teams (2-4 students) to solve P
- Accompanying lectures+soft skill sessions

Teamwork Across Disciplines: Concept



- Restriction to two disciplines $D1$, $D2$
- Design interdisciplinary problem P of (overlapping) sub-problems P_{D1}, P_{D2}
- Form mixed $D1/D2$ student teams (2-4 students) to solve P
- Accompanying lectures+soft skill sessions
- Supervised team meetings (STM)
  - → fixed point of contact for each team
  - → fixed PoC for all teams
    - → global discussions
    - → status reports, ...
  - → fixed PoC of teams and teachers
    - → support teams
    - → detect teamwork issues, ...

Teamwork Across Disciplines: Concept



- Restriction to two disciplines $D1$, $D2$
- Design interdisciplinary problem P of (overlapping) sub-problems P_{D1}, P_{D2}
- Form mixed $D1$/$D2$ student teams (2-4 students) to solve P
- Accompanying lectures+soft skill sessions
- Supervised team meetings (STM)
  - → fixed point of contact for each team
  - → fixed PoC for all teams
    - → global discussions
    - → status reports, ...
  - → fixed PoC of teams and teachers
    - → support teams
    - → detect teamwork issues, ...
- Embed course in curricula of both disciplines $D1$, $D2$

Turbulent Flow Simulation on HPC-Systems: Concept

- $D1$: Computer Science (CS), $D2$: Mechanical Engineering (ME)
- Curriculum: 2×90 min. per week (lecture/soft skills/STM)
- Given: MPI-parallel software package NS-EOF for laminar incompressible fluid flow simulations
  P: extension simulation software NS-EOF to turbulent flows
- Is this “interdisciplinary”?

Turbulent Flow Simulation on HPC-Systems: Concept

- *D*1: Computer Science (CS), *D*2: Mechanical Engineering (ME)
- Curriculum: 2×90 min. per week (lecture/soft skills/STM)
- Given: MPI-parallel software package NS-EOF for laminar incompressible fluid flow simulations
  P: extension simulation software NS-EOF to turbulent flows
- Is this “interdisciplinary”?
  - ✓**C1** Knowledge on turbulence modeling (ME) *and* MPI parallelisation (CS) required
  - ✓**C2** Students/teachers from ME and CS need to be/are involved
  - ✓**C3** Master’s course: prel. knowledge on programming/(comp.) fluid dynamics for both ME/CS

Turbulent Flow Simulation on HPC-Systems: Course Structure

I Single Phase

II Group Phase: Interdisciplinary Teamwork

III Project Phase

Turbulent Flow Simulation on HPC-Systems: Course Structure

I Single Phase

→ students work individually

Duration	2 weeks
Lectures	Basics: (Comp) Fluid Dynamics, C++-Programming, NS-EOF
Outcome	Getting familiar with NS-EOF, Building common ground for teamwork

II Group Phase: Interdisciplinary Teamwork

III Project Phase

Turbulent Flow Simulation on HPC-Systems: Course Structure

I Single Phase

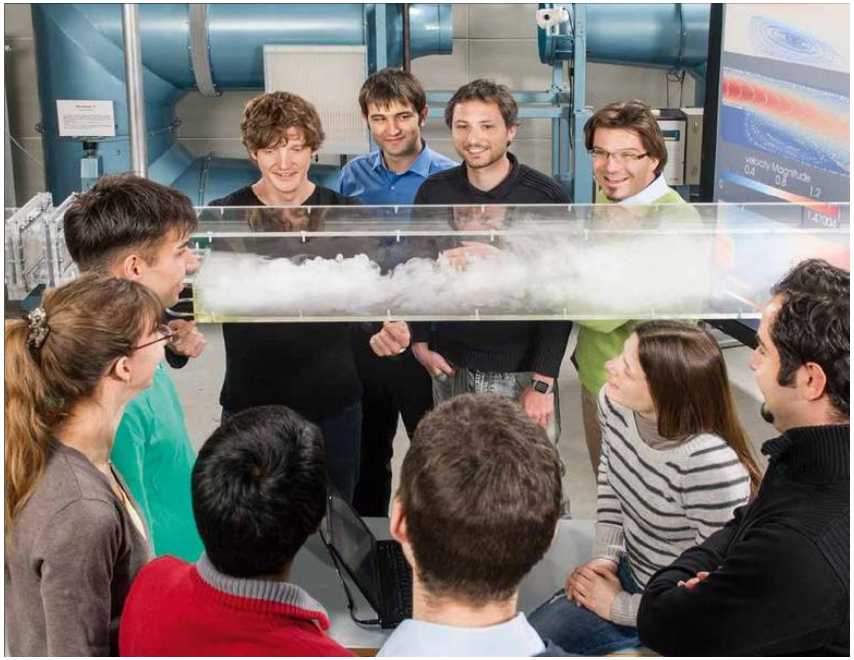
II Group Phase: Interdisciplinary Teamwork

$\rightarrow$ students work in teams

Duration	6 weeks
Lectures	Parallelisation, Turbulence modeling
Soft skills	Teamwork, communication
Outcome	Parallel NS-EOF with turbulence model Teamwork experience and practice

III Project Phase

Turbulent Flow Simulation on HPC-Systems: Course Structure



I Single Phase
II Group Phase: Interdisciplinary Teamwork
III Project Phase

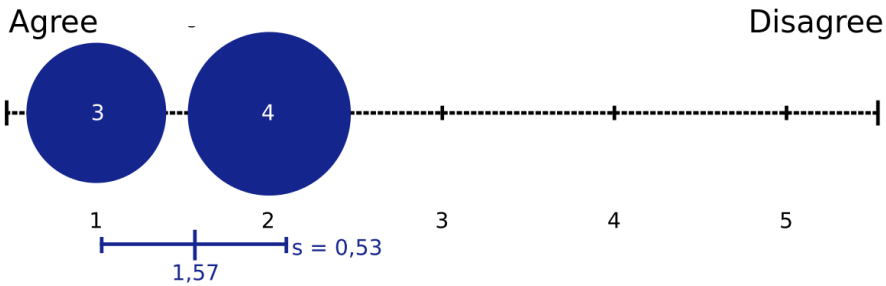
→ students work individually or in teams

Duration	4 weeks
Lectures	Experimental fluid dynamics and wind tunnel tour
Outcome	Individual software extensions/enhancements

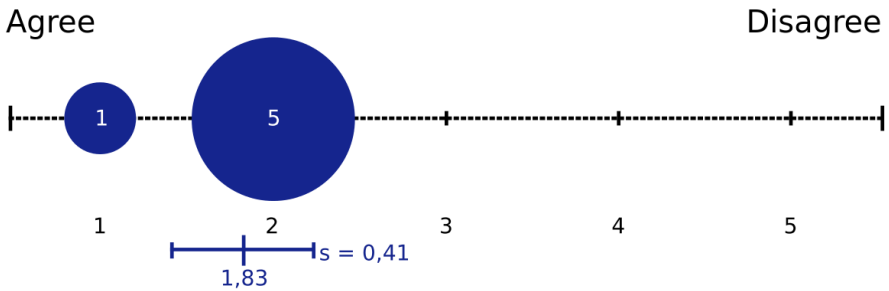
Evaluation: Methods

- “Standard” course evaluation at week 8
- Teaching analysis poll after the course
- Individual feedback discussions with students/student groups during/after course

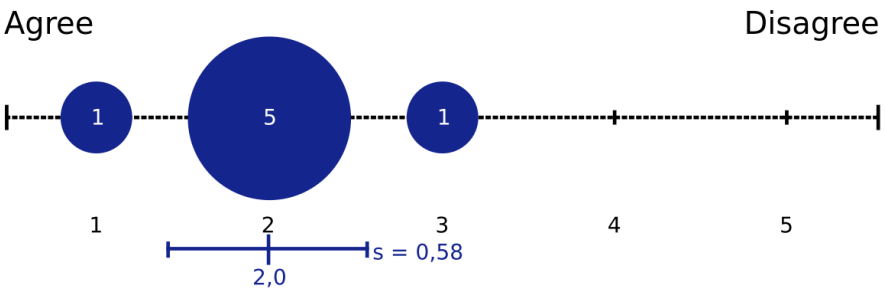
Course Evaluation ($n = 7$)



I am able to successfully work on a typical/ comparable task of this course.



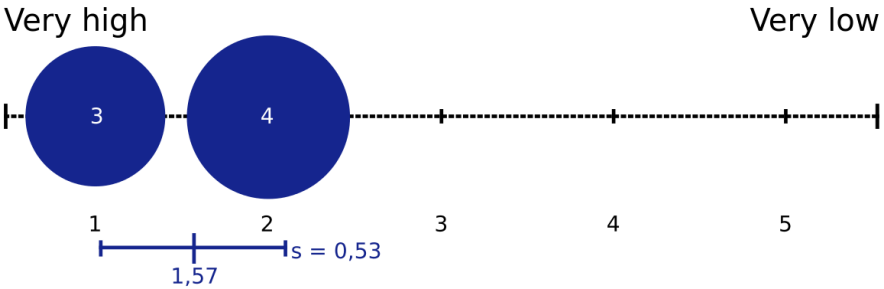
For a certain problem, I can select appropriate tools and methods from computer science.



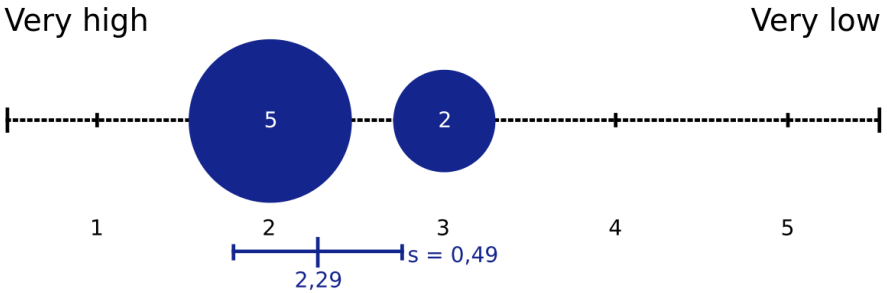
I can apply scientific methods from computer science to concrete tasks.

- CS→ME view: ✓
- Levels “Analysis” and “Application” reached

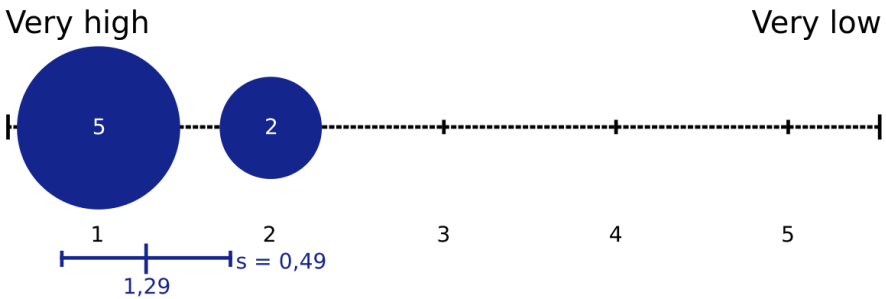
Course Evaluation ($n = 7$)



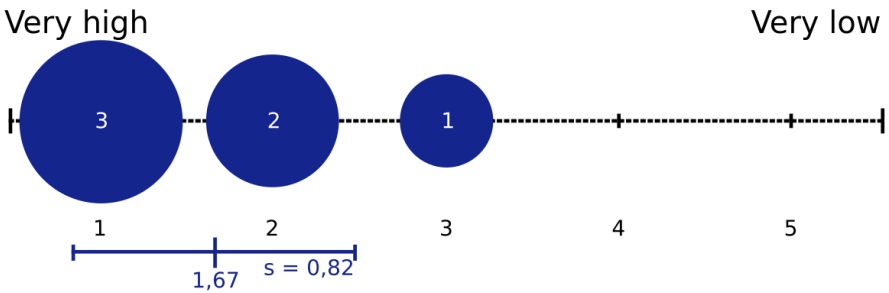
The amount of expected preliminary knowledge is ...



The speed of addressing/discussing the different tasks is ...



The level of difficulty of the tasks/assignments is...



The extent of the different tasks is ...

- Do more, get more ;-)
- $\rightarrow$ revise/improve requisites: enhanced introductions

Evaluation: Excerpt of Teaching Poll Analysis ($n = 13$)

✓ What is most supportive:

- Team/ cooperative work (13)
- Support by supervisors/reachability (13)
- Learning by doing (11)

× What prevents from successful learning:

- No feedback after review sessions (11)
- Time/other lectures (11)
- Too little code documentation (8)

Summary & Conclusion

- Methodology *Teamwork Across Disciplines* to enable interdisciplinary, supervised team-based learning
- Exemplary course: HPC meets Engineering
- Evaluation: team-based learning supports acquisition of competences
- In future: other “example courses”?

Summary & Conclusion

- Methodology *Teamwork Across Disciplines* to enable interdisciplinary, supervised team-based learning
- Exemplary course: HPC meets Engineering
- Evaluation: team-based learning supports acquisition of competences
- In future: other “example courses”?

Thank you for your attention!

This work was supported by *TUM ProLehre* and *TUM Hochschulreferat Studium und Lehre*. We thank the Munich Centre of Advanced Computing (MAC) for providing computational resources. Financial support by the DFG priority program *1648 Software for Exascale Computing* is acknowledged.