



Implementing the Software Development trajectory of the Computer & Embedded-Systems Engineering's masters


CURRICULUM 2020/2021 – old school 

- set of independent courses
- variety of programming languages, tools, methods
- mixed intake of students with no differentiation
- limited formative feedback
- grading by black box testing

WANTED – student-centered program

- homologation for BSc EE intake
- robust systems programming language (Rust)
- model-based development
- agile workflow (Git)
- group work (scrum)
- formative feedback (code quality)
- automated grading (efficiency, fairness)

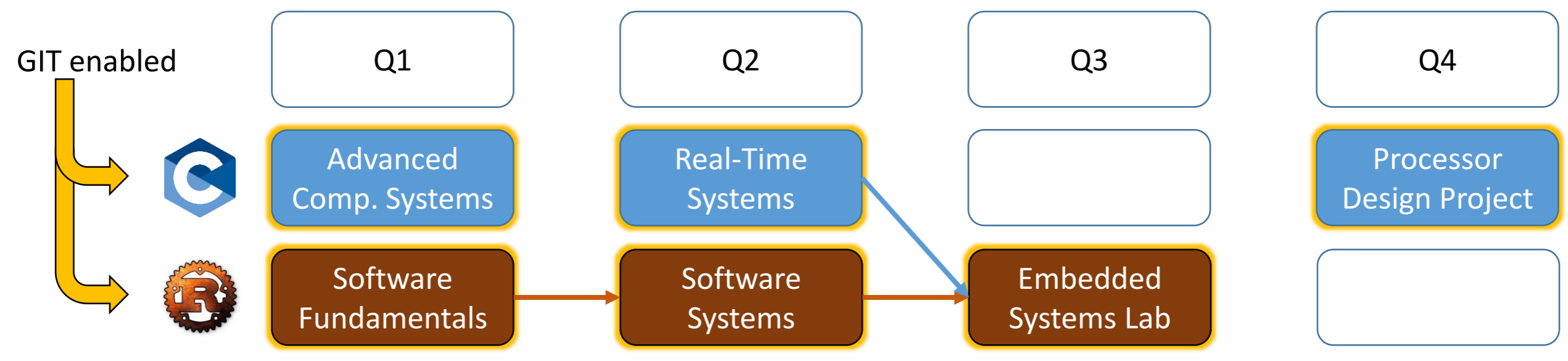


CURRICULUM COMMITTEE (2021) 

- merge CE and ES master programs
- educate for the future
- learning trajectories
 - computer architecture (CE)
 - embedded software development (ES)
 - skills (team work, effective & responsible eng.)
- program vs teacher-based organization (coherence)

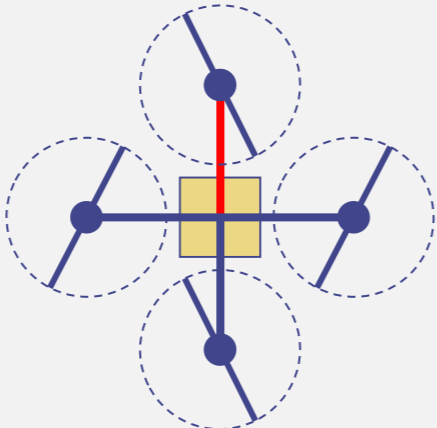
TEACHING TEAM XS

- automate tasks to enhance efficiency
- in-house development of support tools
 - import/export interface to BrightSpace portal
 - content management with versioning
 - code templates per lab



DIDACTIC CONCEPTS

- learning by doing
- practice makes perfect
- repetition with variation



SOFTWARE DEVELOPMENT TRAJECTORY

- projects (small, medium, large)
- self study – weblab tool (formative feedback)
- group work
 - size 2, 4
 - different composition
 - extensive TA support (formative feck)
- HW in the loop
 - synthesizer
 - quadcopter

CURRICULUM 2023/2024

- homologation with Rust + Git
 - self study + grp work
 - weblab (autom. grading & feedback)
- softw. devel. trajectory spanning 3 courses
 - **Rust** programming language
 - set of TAs following along
- **Git** tooling supporting 6 courses
 - agile development
 - automated grading
- model-based development
 - UML
 - State charts
 - Domain Specific Languages

