UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

# Whitepaper: Project-based curricula

van Albada, D.; Bakker, R.; Bethke, I.; Belleman, R.; van den Berg, D.; Bruntink, M.; Dekkers, H.; Douma, R.; van Inge, T.; Lagerberg, J.; Pimentel, A.; Polstra, S.; Poss, R.; Varbanescu, A.L.; Visser, A.; Zaytsev, V.

**Publication date**
2014

**Document Version**
Final published version

Link to publication

**Citation for published version (APA):**
van Albada, D., Bakker, R., Bethke, I., Belleman, R., van den Berg, D., Bruntink, M., Dekkers, H., Douma, R., van Inge, T., Lagerberg, J., Pimentel, A., Polstra, S., Poss, R., Varbanescu, A. L., Visser, A., & Zaytsev, V. (2014). *Whitepaper: Project-based curricula*. Universiteit van Amsterdam.

**General rights**
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

*UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)*

Download date:10 nov 2026

# Whitepaper: project-based curricula

**Institute for Informatics: Dick van Albada, Roy Bakker, Inge Bethke, Robert Belleman, Daan van den Berg, Magiel Bruntink, Hans Dekkers, Roeland Douma, Toto van Inge, José Lagerberg, Andy Pimentel, Simon Polstra, Raphael Poss, Ana-Lucia Varbanescu, Arnoud Visser, Vadim Zaytsev**

**October 7th, 2014**

On June 17th, 2014, a colloquium took place between members of the teaching staff of the Institute for Informatics of the University of Amsterdam. The goal of this colloquium was to explore new or different ways to structure teaching in Informatics (Computer Science) using student projects.

The colloquium was opened by a presentation from dr. Akim Demaille from EPITA, a technical university in Paris, France. During his presentation, dr. Demaille shared his experience using Tiger, a 6-month project given to BSc-3 students, around which multiple courses are articulated, including: formal languages, automata theory, software modeling, object-oriented programming. The topic of the project is the construction of a compiler for the Tiger language, based on the book "Compiler Construction" by A.W. Appel. However, as dr Demaille insisted, the goal of the project is not to construct a compiler; instead, students are graded based on their ability to apply the knowledge gained during the related courses, *as well as their overall mastery of software engineering*: how to develop tests for regression testing, how to effectively debug faults, how well tasks are divided among the peer group, etc. This teaching activity has been running since 2000.

The second part of the colloquium was organized in focus groups, to consider whether and how new or different project-centered activities could be taken on board the current Informatics curricula. The rest of this whitepaper summarizes the findings of this discussion.

## Opportunities

The major finding of the colloquium was the existence of a strong shared interest among the teaching staff for more project-based teaching activities. Multiple **motivations** for this were phrased:

- several courses, including logic, automata theory or algorithms & complexity seem excessively difficult to grasp by students. The teachers posit that student **miss a "big picture"** that connect these courses to other themes in computer science;
- in the current curricula, practical assignments are graded based only on how well the students apply the course's knowledge. In particular, subsequent MSc courses usually assume that students teach themselves the **engineering side of programming**: using version control, debuggers, etc., however in practice the resulting skills are too heterogeneous;
- there exists a cluster of teachers in the core Informatics curriculum who wish to coordinate their courses towards a **layered introduction to Computer Systems**: from gates to circuits, from circuits to processors, from instruction sets to operating systems, etc., and thus exploit a common backbone of coordinated student activities.

When requested to identify or devise ways to integrate new or different project-based activities in their work, the colloquium members phrased multiple **strategies**:

- **Stable individual refactoring**: the portfolio remains unchanged; each teacher develops individually new activities within their course that promote project-based work by students;
- **Stable coordinated refactoring**: the portfolio remains unchanged; teachers develop collectively a common set of project-related skills and criteria that are subsequently tested in each course individually using different course-specific projects;
- **Extrinsic coordination**: the portfolio is extended with one or more "project course", which anchors applications from one or more other courses onto a common project timeline, and where students can earn distinct credits (eg. Maastricht science program);

- **Organic integration**: the portfolio remains unchanged; teachers develop collectively a common project split into “modules”, with each course “contributing” its distinguishing module. Project credits are earned by module, as part of the existing course's credits.

A large consensus was established that stable refactoring, either individual or coordinated, is desirable regardless of circumstances and is to be undertaken “in the field” as a matter of course. Some teachers highlighted this process is already ongoing. Extrinsic coordination was the first large-scale proposal directly resulting from the opening by dr. Demaille, but was met with specific criticism detailed in “Pitfalls” below. **Organic integration was identified as the alternative approach that answers the motivations while avoiding the pitfalls.** The attendees agreed to set up working groups to elaborate on this opportunity later.

**Pitfalls**

The following pitfalls were identified during the colloquium:

- **undesirable dependencies**. When a large project mandates a separation into a fixed order of steps, like with the Tiger compiler, the corresponding chaining of related courses is a new constraint introduced for the students that would not exist otherwise. This in turn prevents students from following these courses in a different order, or starting their study “in the middle” of a course sequence, re-taking individual courses, etc. This should be deemed undesirable, as long as flexibility remains a strong selling point of the education program;
- **project topics and education level.** While it is tempting to let “motivated” students come up with their own project topics, student motivation is not sufficient for high quality learning. The teaching staff should continue to “own” project topics in order to guarantee the right level of education, especially towards renewal of the program's accreditation;
- **experience of the teaching staff with large projects.** If/when a teaching team agrees to extend their program with an overarching student project that spans multiple courses, the risk exists that the first few iterations of this activity will fail to achieve all desired teaching objectives due to lack of experience by the teaching staff. This risk must be adequately mitigated, so as to avoid “wasting” the time and financial investment of students participating in these first iterations.

**Modalities**

Consensus was reached to acknowledge that large groups of students are unavoidably heterogeneous and *projects thus need to cater for different student levels.* The Tiger project's separation in a “standard” part for all students and one or more “professional” or “expert” parts eligible for extra credits was found to be an attractive universal approach.
The colloquium also welcomed the confirmation by dr. Demaille's experience report that *student skill levels within a peer group must not be too heterogeneous* (as would happen, for example, by forcing “good” students to pair with “poor” students), especially for groups larger than 2, for otherwise the better students end up doing most of the work and develop a distaste for collaboration.

**Conclusions**

Clear motivations and opportunities have been identified to link existing courses in the UvA portfolio around common student projects. Most Informatics teachers have committed to self-organize in this direction, at least by coordinated refactoring. Organic integration, where different courses provide “modules” that can be subsequently reused by an overarching student project, will be also explored.