Open Access Research Journal of
Science and Technology
Journals home page: https://oarjst.com/
ISSN: 2782-9960 (Online)

OARJ OPEN ACCESS RESEARCH JOURNALS

(RESEARCH ARTICLE)

Regulation-anchored competency engineering (RACE): A foundational framework for a living science of cyber-technology education

Philippe Funk [1, 2, *]

[1] *Business Administration, Information Technology and Cybertech, Signum Magnum College (SMC), Portomaso Business Centre, Portomaso, St. Julians PTM 01, Malta.*
[2] *Polytechnic Cyber Institute, 3, rue Nicolas Petit, L-2326 Luxembourg, Grand Duchy of Luxembourg.*

Open Access Research Journal of Science and Technology, 2026, 17(02), 008–017

Publication history: Received on 09 June 2026; revised on 14 July 2026; accepted on 17 July 2026

Article DOI: https://doi.org/10.53022/oarjst.2026.17.2.0069

Abstract

Curricula in cybersecurity, artificial intelligence governance, and data protection begin to age the moment they are approved. A module written against the regulations in force in one academic year is, by its second delivery, teaching a snapshot; by its third, nobody can say precisely which parts have gone stale without rereading everything. I argue that this is not primarily a problem of academic effort or funding, but of modelling. We have no formal representation of what a curriculum is anchored to, so we cannot compute what has changed. This paper proposes regulation-anchored competency engineering (RACE), which treats regulatory and competency obligations as dated, versioned artifacts that curricula are compiled from, rather than as background documents consulted occasionally. RACE adds a fourth vertex to constructive alignment: alongside learning outcomes, teaching activities, and assessment, there is an explicit obligation layer to which all three must be traced. Curricula then become typed property graphs across five layers, namely regulation, competency, outcome, activity and assessment. Three things follow: a drift metric that quantifies curriculum obsolescence, a continuous-integration discipline called CurriculumOps that flags affected courses when an upstream instrument changes, and compilation of syllabi and traceability matrices from the source. Generality is demonstrated through anchor maps for two domains and two complete module compilations, one in cybersecurity risk assessment and one in AI governance. The framework is conceptual and analytically evaluated. It has not yet been piloted, and I set out what would count as evidence for and against it.

Keywords: Curriculum engineering; Competency-based education; Constructive alignment; Regulatory compliance; Cybersecurity education; Artificial intelligence governance

1. Introduction

1.1. The drift problem

Cyber-technology curriculum drift. A cybersecurity module designed carefully in one academic year will, within months, be partly out of date. The NIS 2 Directive [1] acquires implementing guidance; the EU AI Act [2] moves through its phased application dates; the Markets in Crypto-Assets Regulation (MiCA) [3] is supplemented; the Workforce Framework for Cybersecurity (NICE Framework) [4] publishes revised role definitions; the European Skills, Competences, Qualifications, and Occupations classification (ESCO) [5] reworks a skill cluster; and the attack techniques and professional toolchains that the laboratory exercises quietly assume have moved on. None of these are dramatic. It accumulates.

* Corresponding author: Philippe Funk ORCID: https://orcid.org/0000-0003-4949-3814

By the second year of delivery, a course anchored to a single regulatory snapshot is teaching thinking that is two years old, which is often tolerable. By the third year it may not be, and this is where the practical difficulty bites. Working out exactly which learning outcomes, which activities and which assessment rubrics have gone stale is forensic work. It requires someone to read the amended instrument, hold the whole syllabus in mind, and reconstruct a mapping that was never written down in the first place. Few academics have that time, and the ones who do are rarely rewarded for spending it.

The costs of this are distributed, which is why they are easy to ignore. Accreditation panels cannot readily verify currency, so they verify process instead. Employers form the view that graduates are not job-ready without being able to say precisely where the gap lies. Students pay for a qualification calibrated to a regulatory environment they will not actually enter. Curriculum designers spend their meetings arguing about what ought to be taught rather than building the machinery that would keep the answer current.

Why does the drift persist? Three reasons, and they compound. The first is fragmentation: regulations, competency frameworks, learning outcomes, activities, and assessments live in separate documents maintained by separate people, so a single amendment must be traced by hand through every course that might depend on it. Second, almost none of it is machine-tractable. Requirements arrive as natural-language prose with implicit versioning and no formal structure; alignment is asserted once in a validation document and never measured again; and drift is felt but never counted. The third is the absence of any continuous check. A curriculum, once approved, sits still until a periodic review on a three- to five-year cycle or an accreditation visit forces a rewrite. In between, misalignment accrues silently, which is the worst way for it to accrue.

1.2. Constructive alignment and what it leaves out

Constructive alignment, set out by Biggs [6] and developed by Tang [7], has been the default framework for course design in higher education for roughly three decades, and deservedly so. Its central claim is simple and hard to argue with: intended learning outcomes (LOs), teaching and learning activities (TLAs), and assessment tasks (ATs) must line up with one another. Students cannot demonstrate what they were never given an opportunity to practice, and an assessment that measures something other than the stated outcome is measuring the wrong thing. The model is conventionally drawn as a triangle with three vertices.

The triangle guarantees internal coherence. It says nothing about where the outcomes came from or whether they are still correct. A perfectly aligned course can be perfectly aligned to a requirement that was superseded eighteen months ago, and constructive alignment offers no way to notice. This is not a criticism of Biggs' work. It was never the question the model was built to answer. But it does leave the anchor unexamined, and the anchor is precisely the thing that moves.

1.3. Competency frameworks, regulatory drivers, and the gap between them

Meanwhile, a substantial ecosystem has grown to define what graduates ought to know and be able to do: the NICE Framework [4], ESCO [5], the European e-Competence Framework (e-CF) [8], the ACM/IEEE computing curricula guidelines [9], and accreditation criteria such as those of ABET [10]. Regulation has gone further and now mandates educational outcomes directly. NIS 2 requires cybersecurity training proportionate to the role and risk [1]. The EU AI Act imposes AI-literacy obligations and governance training, with heavier requirements around high-risk systems [2]. The Digital Operational Resilience Act (DORA) requires financial entities to satisfy themselves that information and communication technology (ICT) risk-management competencies are actually present [11]. MiCA attaches conduct-related educational duties to crypto-asset service providers [3]. ISO/IEC 42001 asks organizations to assess and develop AI-governance competence [12]. All of these are living instruments whose implementation guidance keeps arriving, so a curriculum designed against a snapshot starts drifting on the day it is signed off.

There is, then, no shortage of material. What is missing is the join. Alignment theory, competency frameworks and regulatory obligations exist as three separate literatures and three separate document sets, and nothing formally connects them. Requirements are published as prose. Learning outcomes appear in syllabi that are frequently vague, rarely measurable, and almost never carry an explicit pointer to whatever they are meant to satisfy. Traceability from regulation through to assessment is claimed in validation paperwork and, so far as I have been able to establish, never verified computationally. No standard tooling exists for the version control, continuous integration or automated compliance checking of curricula. This is a striking absence, given that software engineering has had all three for well over a decade, and curricula are structurally not so different from a dependency graph.

1.4. Contribution and structure

The argument of this paper is that curriculum drift is a modelling and tooling problem rather than a resource or discipline problem, and that framing it this way makes it tractable. I propose regulation-anchored competency engineering (RACE) as a response. RACE (i) extends constructive alignment from three vertices to four by promoting the regulatory-competency obligation to a first-class, dated, versioned artifact; (ii) formalizes curricula as typed property graphs across five artifact layers; (iii) defines a drift metric that makes obsolescence measurable; (iv) specifies CurriculumOps, a continuous-integration discipline for curricula; and (v) demonstrates domain generality through anchor maps and two end-to-end module compilations.

One caveat belongs here rather than buried in the limitations. RACE is a design artifact evaluated analytically against the problem it was built for. It has not been piloted, and no claim is made about measured learning gains. What is offered is a model, a metric and a workflow that are specific enough to be implemented and therefore specific enough to be wrong. Section 2 sets out the research design and formal model. Section 3 presents the anchor maps and module compilations and discusses implications and limitations. Section 4 concludes.

2. Material and methods

2.1. Research design and materials

The study follows design-science research methodology [13], which produces and evaluates purposeful artifacts (constructs, models, methods and instantiations) addressing identified organizational problems. Four design-science activities structure the work: problem identification, covered in Section 1; artifact design, comprising the four-vertex alignment model, the five-layer artifact graph, the drift metric, and the CurriculumOps workflow, in Sections 2.2 to 2.5; demonstration, through two end-to-end module compilations in Sections 3.2 and 3.3; and analytical evaluation against the requirements derived from the problem analysis, in Sections 3.4 and 3.5.

The materials analysed are publicly available primary sources, taken in the versions in force at the time of writing: the regulatory instruments [1-3, 11], the competency frameworks and classifications [4, 5, 8], the curriculum guidelines and accreditation criteria [9, 10], and the management-system standard [12] identified above. No human or animal subjects were involved, and no primary data were collected. Because the instruments themselves are versioned, the reader should treat the specific article references as anchors current at the stated date rather than as permanent fixtures, which is, of course, the whole point of the paper.

2.2. The four-vertex alignment model

RACE extends constructive alignment by introducing a fourth vertex: the regulatory-competency obligation (RCO), treated as a first-class, dated, versioned artifact rather than as context. An RCO is one specific requirement drawn from a regulation or a competency framework. A worked example: “EU AI Act, Article 10 and Annex III: high-risk AI systems require documented risk-assessment and data-governance training.” The four vertices are therefore the RCO; the LOs, each traceable to one or more RCOs; the TLAs, traceable to LOs; and the ATs, traceable to both LOs and RCOs.

The model asserts an invariant. Every RCO is refined into at least one LO; every LO is operationalized by at least one TLA and assessed by at least one AT. The invariant is deliberately weak because a stronger one would be violated by any real curriculum on its first day. What it buys is this: when an RCO changes, every dependent LO, TLA, and AT is reachable by graph traversal and can be flagged automatically. Alignment stops being a design assertion made once at validation and becomes a property of the curriculum that can be recomputed on demand. That shift, from assertion to recomputable property, is the substantive move in the paper; everything else follows from it.

2.3. The five-layer artifact graph

Curricula are formalized as typed property graphs. Nodes span five artifact layers and edges encode traceability. Table 1 summarizes the layers, their typical sources, and their key properties. Bloom-level properties follow the revised taxonomy of educational objectives [14].

Table 1 The five-layer artifact graph of the RACE framework

Layer	Artifact type	Typical source	Example	Key properties
0	Regulatory artifact (RA)	Regulations; framework releases	NIS 2, Article 21; EU AI Act, Annex III	Version; effective date; jurisdiction; artifact identifier
1	Competency descriptor (CD)	e-CF; ESCO; NICE; domain frameworks	"Conduct risk assessment for AI systems"	Competency identifier; parent competency; proficiency level; mapping to RA; version; last updated
2	Learning outcome (LO)	Course and module syllabi	"Design and execute a risk assessment for a high-risk AI system in accordance with EU AI Act Article 10"	Bloom level; mapping to CD; course identifier; effective term; verifiability
3	Teaching and learning activity (TLA)	Course-design documents	"Laboratory exercise: risk-assessment report for a facial-recognition system"	Duration; modality; mapping to LO; prerequisite knowledge; resource requirements
4	Assessment task (AT)	Rubrics; examination specifications; assignment briefs	"Rubric: correctness of risk-assessment report against ISO/IEC guidelines"	Rubric identifier; mappings to LO and RCO; assessment type; pass threshold

Cross-layer edges carry the traceability semantics: a regulatory artifact governs one or more competency descriptors; a competency descriptor is refined into one or more learning outcomes; a learning outcome is operationalized by teaching activities and assessed by assessment tasks; and an assessment task validates the regulatory requirement from which it ultimately derives. Because every edge is typed, dated, and versioned, the graph supports impact analysis, drift detection, and compilation. The choice of a property graph over a relational schema is pragmatic rather than principled: the queries that matter here are reachability queries, and those are what graphs are good at.

2.4. The drift metric

Curriculum drift is the degree of misalignment between the current state of the curriculum graph and the current state of the regulatory and competency layers. Let R(t) be the set of RCOs effective at time t, C(t) the set of competency descriptors effective at time t, and G(t) the curriculum graph (LOs, TLAs, ATs, and their mappings) at time t. Drift is then:

$$D(t) = [\ |\Delta R(t)| + |\Delta C(t)|\ ] \ / \ |G(t)| \times 100\%$$

where $\Delta R(t)$ is the set of RCOs added or modified since the last curriculum update, $\Delta C(t)$ the set of competency descriptors added or modified since that update, and $|G(t)|$ the total number of nodes in the curriculum graph. Interpretation is banded. $D = 0$ indicates alignment with the instruments currently in force. Values below 10% indicate minimal drift that routine maintenance absorbs. Between 10% and 30%, drift warrants targeted revision. At 30% and above, the module needs a major review. When D crosses a configured threshold, the system alerts the curriculum owner with the specifics: which RCOs changed, which LOs are affected, which courses need review, and an estimate of the remediation effort.

Two honest observations about this metric. It is a ratio of counts, so it treats a trivial editorial amendment and a substantive new obligation as equal, which they plainly are not; a weighted variant is an obvious refinement and is left to future work. And the band thresholds are calibrated by judgement, not by evidence. They are offered as a starting point for institutions to tune, and the first pilot that produces real threshold data should be expected to move them.

2.5. CurriculumOps: continuous integration for curricula

CurriculumOps borrows directly from DevOps and continuous-delivery practice [15] and treats curriculum artifacts as code: version-controlled, automatically tested, continuously delivered. The pipeline has five phases.

- Phase 1: Source control. Regulatory artifacts, competency descriptors, learning outcomes, activities, and assessments are stored in version-controlled repositories. Each artifact is versioned independently, cross-

references use semantic versioning, and every commit carries metadata: author, date, rationale, and the downstream artifacts affected.

- Phase 2: Validation and traceability checking. Automated validators verify that every LO traces to at least one RCO or competency descriptor, every TLA to at least one LO, and every AT to at least one LO and one RCO; that the traceability graph is acyclic; that the Bloom levels of outcomes, activities, and assessments are mutually consistent; and that rubrics align with the outcomes they claim to measure.
- Phase 3: Drift detection and compliance checking. When an upstream regulation or framework is updated, the system identifies the affected competency descriptors, flags the dependent learning outcomes, marks the courses containing them for review, computes the drift metric, and produces an impact report listing affected outcomes, flagged courses, and estimated review time.
- Phase 4: Curriculum compilation. A compiler ingests the regulatory-competency requirements, outcome specifications, activity and assessment designs, and audience parameters (proficiency level, duration, modality). It emits a machine-readable curriculum specification (JSON Schema or RDF, for instance), a human-readable syllabus, a full traceability matrix running from RCO through competency descriptor and learning outcome to activity and assessment, and deployment recommendations.
- Phase 5: Continuous verification and improvement. After deployment, learning outcome attainment, anonymized feedback, and detected alignment violations are logged and fed back into the curriculum graph. This phase is the least developed of the five and the most likely to need rethinking once real attainment data arrive.

A reference toolchain follows straightforwardly: a version-control repository for all artifacts; schema validators (JSON Schema, SHACL, or OWL) for structural and cross-layer constraints; a traceability analyser built on graph traversal; a drift calculator with alerting; a curriculum compiler realized as a domain-specific language with code generation to syllabi and traceability matrices; continuous-integration pipelines running validation and drift checks on every update; and a dashboard reporting curriculum status. Nothing in this list requires invention. Every component exists as a mature technology in software engineering; what is missing is the will to apply it to curricula.

3. Results and discussion

3.1. Domain-specific anchor maps

RACE is domain-agnostic, but it is only as good as its anchor mapping, which is the work of establishing the correspondence between regulations, competency frameworks, and pedagogical outcomes within a given domain. This is where the intellectual labour actually sits, and no amount of tooling removes it. Tables 2 and 3 present the anchor maps produced for cybersecurity and AI governance. Their primary drivers are, respectively, the NIS 2 Directive [1], the NIST Cybersecurity Framework [16], and the NICE Framework [4]; and the EU AI Act [2], ISO/IEC 42001 [12], and, for AI systems in the financial sector, DORA [11].

Table 2 Key regulatory-competency anchors for the cybersecurity domain

RCO ID	Source	Description	Proficiency	Mapped roles (NICE)
SEC-001	NIS 2, Art. 21 [1]	Provide cybersecurity awareness training	Foundation	All roles
SEC-002	NICE, Operate and Maintain [4]	Manage security patches and updates	Advanced	System Administrator; IT Operations
SEC-003	NICE, Oversee and Govern [4]	Conduct security risk assessments	Expert	Security Manager; Risk Officer
SEC-004	NIST CSF [16]	Implement access-control mechanisms	Advanced	Access-Control Analyst; IAM Specialist

Representative learning outcomes compiled from these anchors include: explaining the business impact of a phishing attack and recommending user actions consistent with NIS 2 awareness requirements (SEC-001); designing and executing a patch-management plan for a 500-person organization with documented NICE competency coverage (SEC-002); and conducting a risk assessment of a legacy information technology (IT) system and identifying regulatory gaps under NIS 2 (SEC-003).

Table 3 Key regulatory-competency anchors for the AI-governance domain

RCO ID	Source	Description	Proficiency	Mapped roles
AI-001	EU AI Act, Annex III [2]	Conduct risk assessment for high-risk AI systems	Expert	AI Risk Officer; Governance Lead
AI-002	EU AI Act, Art. 10 [2]	Document training-data provenance and bias testing	Advanced	AI Practitioner; Data Scientist
AI-003	ISO/IEC 42001 [12]	Establish AI-governance framework and accountability	Expert	AI Governance Officer; CTO
AI-004	DORA, Art. 17 [11]	Monitor and report AI-system performance anomalies	Advanced	AI Systems Operator; QA Lead

Representative learning outcomes here include: evaluating a proposed high-risk AI system against EU AI Act criteria and identifying compliance gaps (AI-001); documenting the training dataset of a credit-scoring system, including bias-testing results and mitigation measures (AI-002); and designing an organizational AI-governance framework covering accountability, monitoring and third-party auditing in line with ISO/IEC 42001 (AI-003).

The two maps are structurally similar despite covering different domains, which is the point of including both. The anchors differ, the roles differ and the proficiency distributions differ, but the shape of the mapping does not. This is weak evidence for generality rather than strong evidence, and I would not want to claim more from two domains than two domains can support.

3.2. End-to-end demonstration 1: cybersecurity risk assessment

Table 4 presents the first module compilation. CYB-401 is an advanced module on cybersecurity risk assessment anchored to SEC-003, NIS 2 Article 21 [1], and the NICE Security Manager role [4]. Its risk assessment methodologies follow NIST Special Publication 800-30 [17] and IEC 31010 [18].

Table 4 Compiled module CYB-401: cybersecurity risk assessment (advanced level)

Element	Specification
Module identifier; duration; audience	CYB-401; one semester (12 weeks, 3 h per week); junior cybersecurity professionals and IT managers transitioning to security roles
Regulatory-competency anchors	SEC-003 (conduct security risk assessments, expert level); NIS 2 Directive, Article 21 [1]; NICE role: Security Manager, Oversee and Govern [4]
Learning outcomes (Bloom level; anchors)	LO1 (evaluate; SEC-003, NIS 2 Art. 21): Critically evaluate an organization's security posture against the NIST CSF [16] and NIS 2 requirements, identifying gaps, and remediation priorities. LO2 (analyse; SEC-003, NICE): analyse threat models, asset criticality and control effectiveness to estimate risk to business operations. LO3 (apply; SEC-003): apply risk assessment methodologies (NIST SP 800-30 [17]; IEC 31010 [18]) to a case organization. LO4 (analyse; SEC-003): analyse the effectiveness of mitigation strategies and justify trade-offs among security, cost and operational impact
Teaching and learning activities	TLA1 (weeks 1-2): Lectures and facilitated discussions on the NIST CSF, NIS 2 obligations, and governance structures. TLA2 (weeks 3-4): case-study analysis of published incident reports and regulatory findings. TLA3 (weeks 5-8): hands-on laboratory, risk assessment of a sanitized organizational scenario using IEC 31010, documented in a formal report. TLA4 (weeks 9-10): peer review and expert feedback on draft reports. TLA5 (weeks 11-12): practitioner guest lecture and reflective discussion of implementation challenges
Assessment tasks	AT1 (formative, week 4): quiz on the NIST CSF, NIS 2 requirements and risk terminology; pass threshold 75%. AT2 (formative, week 8): rubric-based evaluation of the draft risk-assessment

	report, covering comprehensiveness of threat identification (25%), accuracy of asset valuation and criticality (25%), appropriateness of control evaluation (25%), and clarity and actionability of recommendations (25%); pass threshold 70%. AT3 (summative, week 12): final report and 20-minute panel presentation; same rubric plus professionalism of presentation; pass threshold 75%

Drift monitoring for this module runs on two trackers. The RCO tracker flags CYB-401 if NIS 2 is amended, for instance if the training requirements in Article 21 are revised. The competency-descriptor tracker re-evaluates the mapping of LO1 through LO4 whenever the NICE Framework publishes new Security Manager competencies. Drift is computed quarterly, and if D exceeds 15% the module owner receives an alert of the form: two regulatory anchors have been updated, estimated review time eight hours. The estimate is the weakest part of the alert, since effort estimation for curriculum revision has no established basis, but an approximate figure is more useful to a module owner than none.

3.3. End-to-end demonstration 2: AI governance and regulatory compliance

Table 5 presents the second compilation. AI-410 is an advanced module for AI practitioners, data scientists and governance professionals in regulated industries, anchored to AI-001 and AI-003, EU AI Act Article 10 and Annex III [2], ISO/IEC 42001 [12] and, in the financial-sector variant, DORA Article 17 [11].

Table 5 Compiled module AI-410: AI governance and regulatory compliance (advanced level)

Element	Specification
Module identifier; duration; audience	AI-410; one semester (14 weeks, 4 h per week); AI practitioners, data scientists and governance professionals in regulated industries (financial services, healthcare, autonomous systems)
Regulatory-competency anchors	AI-001 (risk assessment for high-risk AI systems, expert level); AI-003 (AI-governance framework, expert level); EU AI Act, Article 10 and Annex III [2]; ISO/IEC 42001, governance clauses [12]; DORA, Article 17 [11]
Learning outcomes (Bloom level; anchors)	LO1 (evaluate; AI-001): evaluate the regulatory obligations of a proposed high-risk AI system under the EU AI Act and design a compliance roadmap. LO2 (analyse; AI-001, AI-002): analyse high-risk AI-system documentation (design, training data, testing) against EU AI Act requirements and identify deficiencies. LO3 (analyse; AI-003): analyse organizational AI-governance structures and recommend changes to meet ISO/IEC 42001 requirements. LO4 (apply; AI-002): apply bias-detection and mitigation techniques to a training dataset and document results per EU AI Act Article 10. LO5 (apply; AI-004): apply performance monitoring and anomaly detection to a deployed AI system and design intervention protocols per DORA Article 17
Teaching and learning activities	TLA1 (weeks 1-2): lectures and seminar on EU AI Act structure, risk classification and ISO/IEC 42001. TLA2 (weeks 3-4): guest practitioners (compliance officer, data scientist, AI-ethics lead) on real-world governance challenges. TLA3 (weeks 5-7): dissection of published compliance documentation and third-party audit reports. TLA4 (weeks 8-11): team project, a full compliance assessment and governance design for a realistic high-risk AI system such as credit scoring, hiring, or autonomous-vehicle decision logic. TLA5 (weeks 12-13): peer review and external reviewer feedback. TLA6 (week 14): project presentations to a panel with industry and regulatory perspectives
Assessment tasks	AT1 (formative, week 4): examination on EU AI Act risk classification, ISO/IEC 42001 and DORA requirements; pass threshold 75%. AT2 (formative, week 7): case-study analysis report (2,000 words) on a published AI system and its regulatory alignment; rubric covering accuracy, depth and clarity; pass threshold 70%. AT3 (summative, week 13): team project submission comprising system description and risk classification, compliance gap analysis, ISO/IEC 42001 governance design, bias-testing and mitigation strategy, and monitoring and incident-response plan; rubric per section: completeness (70%), regulatory accuracy (20%), feasibility and clarity (10%); pass threshold 75%. AT4 (summative, week 14): 30-minute team presentation with panel questioning; rubric covering clarity, depth and responsiveness; pass threshold 75%

The RCO tracker for AI-410 monitors amendments and implementing acts of the EU AI Act and, once an act is finalized, generates a diff report: Article 10 amended, three requirements changed, two new sub-requirements added, recommended review twelve hours. The competency-descriptor tracker re-evaluates LO3 whenever ISO/IEC 42001 is revised. Drift is computed quarterly with an alert threshold of D above 20%, set higher than CYB-401's because the AI Act is still in its phased application period and a lower threshold would generate alerts faster than anyone could act on them. Threshold selection, it turns out, is a governance decision dressed up as a technical one.

3.4. Implications

Accreditation. Traditional accreditation verifies programmes against static standards at intervals, typically of six years. RACE makes continuous accreditation possible: accreditors define competency-regulatory baselines, programmes implement curricula as RACE graphs with continuous drift monitoring, and accreditors ingest those graphs and verify alignment automatically. The accreditation visit then changes character, from a document audit to a system audit. The question becomes whether the curriculum system is sound, whether the drift thresholds are defensible, and whether the alerts are actually being acted on, rather than whether a particular folder of paperwork was complete on a particular day. Whether accreditation bodies would welcome that shift is a separate question, and I suspect the answer varies considerably by jurisdiction.

Programme quality and student outcomes. Students would be taught against the requirements in force at their graduation date rather than at the curriculum design date two or three years earlier. Employers could verify that programmes map to recognized framework standards [4, 5, 8] instead of inferring it from module titles. Outcome and feedback data would flow back into the curriculum graph and support evidence-based refinement. And verified outcome sets that map to recognized competencies support micro-credentials and stackable certifications almost for free, since the mapping work has already been done.

Curriculum engineering as a discipline. RACE treats curriculum design as an engineering problem, with specifications, formal semantics, version control, automated testing, continuous integration and measured quality. Curriculum designers would become curriculum engineers, trained in formal methods, graph theory and continuous-integration practice. Curricula would become compiled artifacts subject to automated verification, and programme quality would be measured through drift, traceability coverage and alignment metrics rather than asserted. As institutions accumulate artifacts and historical data, they build durable knowledge about which activities actually work for which outcomes, which is knowledge that currently leaves with the individual who retires.

Interoperability. Once curricula are expressed as formal graphs with explicit mappings to regulations and frameworks, they become interoperable. Institutions could compose curricula from versioned, traced libraries of outcomes and activities. Employers could specify sought competencies and search curricula against them. Accreditors and regulators could run automated compliance audits. Aggregate graph data would enable research on pedagogical effectiveness, competency evolution and skills forecasting. This is the most speculative of the four implications and depends on adoption reaching a scale that no single institution can produce.

3.5. Limitations

The most important limitation is the one already stated: RACE has not been piloted. Everything above is a design argument, and design arguments have a well-known tendency to survive contact with reality less well than their authors expect. Beyond that, five qualifications apply.

Adoption cost is real and front-loaded. RACE requires investment in tooling, training and process change, and early adopters bear costs that later adopters will not. Regulatory dynamics cut against the framework's central assumption: RACE presupposes that regulations can be versioned and precisely specified, whereas in practice ambiguity and interpretation disputes persist, and some instruments remain in flux for years. Cultural change may be the binding constraint. Treating curricula as code departs sharply from academic tradition, and faculty may reasonably read the formalization as an encroachment on pedagogical autonomy; if they do, no amount of tooling will save it.

Outcome specification is a genuine difficulty rather than a manageable one. Constructive alignment assumes crisp, measurable outcomes, but many of the outcomes we care about most, such as critical thinking and professional judgement, are irreducibly fuzzy. RACE does not resolve this. It makes the fuzziness visible, which is worth something but is not a solution, and there is a real risk that formalization quietly privileges what is easy to trace over what matters. Finally, scalability and governance: maintaining anchor mappings across hundreds of frameworks and thousands of provisions needs infrastructure and, more awkwardly, an answer to the question of who maintains the master registry and who pays for it. I do not have one.

4. Conclusion

Cyber-technology curricula drift away from the regulatory and competency requirements they are supposed to serve, and the reason is that we have never modelled the relationship well enough to compute it. Regulation-anchored competency engineering is an attempt to do so: it extends constructive alignment to a four-vertex model anchored in explicit regulatory-competency obligations, formalizes curricula as typed property graphs across five artifact layers, defines a drift metric that makes obsolescence a number rather than a feeling, specifies a continuous-integration discipline for curricula, and demonstrates domain generality through anchor maps and two complete module compilations in cybersecurity and AI governance. Under this discipline a course stops being a static document and becomes a compiled, testable, continuously re-verified artifact of a living regulatory-competency codebase.

The obvious next step is to find out whether any of this survives use. Near-term work should pilot RACE in two or three programmes across different domains, build open-source tooling for version control, validation and drift detection, and establish governance structures and registries for anchor mappings. Pedagogical research comparing RACE-designed curricula with conventionally designed ones would be the real test, and it is the test I would most like to see fail informatively. If the framework holds, the longer-term prize is a curriculum-engineering profession, a shorter feedback loop between what industry and regulators require and what institutions teach, and graduates who arrive aligned with the environment they are actually entering rather than the one that existed when their programme was written.

Compliance with ethical standards

Acknowledgments

The author thanks Signum Magnum College (SMC), Malta, and the Polytechnic Cyber Institute, Luxembourg, for institutional support. No external funding was received for this work.

Disclosure of conflict of interest

The author declares no conflict of interest regarding the publication of this manuscript. Philippe Funk: no competing financial or non-financial interests.

References

[1] European Parliament, Council of the European Union. Directive (EU) 2022/2555 of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS 2 Directive). Off J Eur Union. 2022;L333:80-152.

[2] European Parliament, Council of the European Union. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Off J Eur Union. 2024;L,2024/1689.

[3] European Parliament, Council of the European Union. Regulation (EU) 2023/1114 of 31 May 2023 on markets in crypto-assets (MiCA). Off J Eur Union. 2023;L150:40-205.

[4] Petersen R, Santos D, Smith MC, Wetzel KA, Witte G. Workforce framework for cybersecurity (NICE Framework). Gaithersburg (MD): National Institute of Standards and Technology; 2020. NIST Special Publication 800-181, Revision 1.

[5] European Commission. ESCO: European skills, competences, qualifications and occupations classification [Internet]. Brussels: European Commission; 2024 [cited 2026 Jul 10]. Available from: https://esco.ec.europa.eu/

[6] Biggs J. Enhancing teaching through constructive alignment. High Educ. 1996;32(3):347-64.

[7] Biggs J, Tang C. Teaching for quality learning at university: what the student does. 4th ed. Maidenhead: Society for Research into Higher Education and Open University Press; 2011.

[8] European Committee for Standardization. EN 16234-1:2019. e-Competence Framework (e-CF): a common European framework for ICT professionals in all sectors. Part 1: framework. Brussels: CEN; 2019.

[9] ACM/IEEE-CS/AAAI Joint Task Force on Computing Curricula. Computer science curricula 2023 (CS2023). New York: Association for Computing Machinery; 2024.

[10] ABET. Criteria for accrediting computing programs, 2025-2026. Baltimore (MD): ABET; 2024.

[11] European Parliament, Council of the European Union. Regulation (EU) 2022/2554 of 14 December 2022 on digital operational resilience for the financial sector (DORA). Off J Eur Union. 2022;L333:1-79.

[12] International Organization for Standardization, International Electrotechnical Commission. ISO/IEC 42001:2023. Information technology: artificial intelligence: management system. Geneva: ISO; 2023.

[13] Hevner AR, March ST, Park J, Ram S. Design science in information systems research. MIS Q. 2004;28(1):75-105.

[14] Anderson LW, Krathwohl DR, editors. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. New York: Longman; 2001.

[15] Humble J, Farley D. Continuous delivery: reliable software releases through build, test, and deployment automation. Boston (MA): Addison-Wesley; 2010.

[16] National Institute of Standards and Technology. The NIST cybersecurity framework (CSF) 2.0. Gaithersburg (MD): NIST; 2024. NIST CSWP 29.

[17] National Institute of Standards and Technology. Guide for conducting risk assessments. Gaithersburg (MD): NIST; 2012. NIST Special Publication 800-30, Revision 1.

[18] International Electrotechnical Commission. IEC 31010:2019. Risk management: risk assessment techniques. Geneva: IEC; 2019.

Author's short profile

Philippe Funk is affiliated with the Signum Magnum College (SMC), Malta, and with the Polytechnic Cyber Institute, Luxembourg. His research interests include emerging technology education, competency-based curriculum design, regulatory compliance (NIS 2, EU AI Act, DORA), and the application of software-engineering methods to educational quality assurance. ORCID: https://orcid.org/0000-0003-4949-3814