Directing Autonomous Agents New core competency for the MBA graduate

Mikael Alemu Gorsky[1] and Ilya Levin[2]

[1]Holon Institute of Technology, Holon, Israel, mikaelg@hit.ac.il
[2]Holon School of Computer Science, Holon Institute of Technology, Holon, Israel, levini@hit.ac.il

Abstract

Business schools teach students to converse with generative AI. The work those graduates will do is being reshaped by intentional autonomous agents that execute multi-step tasks across files and applications and return finished deliverables. This paper argues that the gap is wide enough to require a change at the level of the curriculum, and that fluency in directing such agents should become a graded competency in the MBA core. The argument draws on the authors' experience teaching the adjacent discipline to software engineers, and it adapts a discipline-based curriculum for agentic software engineering to the deliverable-centred work of the business graduate. The competency is a single analytical judgment, the direction of an agent, with oversight, verification, and the safe handling of data as dimensions of doing that one thing well. Empirical work on knowledge-worker productivity, on the failure modes of agentic systems, and on how students actually use these tools supports the diagnosis and the design. The paper offers a teaching pattern built around a planted error: the agent produces a result that is plausible and wrong, and the student has to catch it. The value a graduate adds lives in the check. This is a conceptual and pedagogical framework grounded in teaching practice, and the paper sets out the classroom it comes from, the limits of what that classroom supports, and the case for carrying a discipline developed with engineering students into the business school. It closes with two models for placing the competency in a programme, the obstacle of faculty readiness, and an admission that the framework awaits empirical validation.

Keywords: *agentic AI, MBA curriculum, autonomous agents, assessment, verification, faculty development.*

1. The training and the workplace have come apart

A business school today can teach a student to use AI well and still send that student into a job for which the training does not prepare them. The lesson and the work have drifted apart, and the drift is recent enough that few faculty have noticed it. Most AI instruction in MBA programmes treat's the technology as something a person talks to. The student learns to write a good prompt, to ask a chat assistant for a summary or a first draft, to push back on a weak answer and ask again. These are real skills. A year ago they were close to the whole of what the workplace asked for.

The workplace has since moved. Tools that now enter business roles do not wait for a prompt and return a paragraph. They accept a goal stated as an outcome, plan the steps that reach it, work across a person's files and applications, and hand back a finished deliverable for the person to check. A junior analyst no longer asks the model how to structure a valuation and then builds it by hand. The analyst gives the model the folder of statements and the instruction to build it, then judges what comes back. What matters has shifted from holding a good conversation to directing a capable, autonomous co-worker. These are different skills.

This paper proposes a way to close that gap. It sets out a framework for teaching agentic AI as a graded competency in the MBA core, and uses Claude Cowork as the concrete instance of the tools graduates will meet, a system whose name signals what it is. An argument about a category is easier to follow with one real example in view. The framework itself depends on no particular tool. It rests on a claim about what a person must know to stand behind work an agent produced, and that claim is the subject of everything that follows. Section 4 sets out the classroom the framework comes from and the limits of what that classroom supports. The framework awaits empirical validation, and cannot receive it until enough programmes adopt the position argued here to produce cohorts worth measuring. A pedagogy has to be proposed and taught before it can be tested. The proposal should not wait for evidence that only adoption can generate.

2. What changed, and what the change is called

The shift has a clean dividing line, and the whole argument for changing the curriculum turns on that line being real, not a matter of degree.

On one side sits the generative model used as a conversational tool. The person supplies a prompt, the model returns text or code or an image, and the person does everything else. Answering is all it does. Orchestration, file handling, and the decision about what to do with the answer all remain with the human. A student who has only ever used AI this way has used a fast source of drafts and has assembled all the work themselves.

On the other side sits the agent, which the graduate will meet less as a tool than as a new kind of co-worker. The person supplies an outcome rather than a step, the autonomous system plans

the path to that outcome, and it carries the plan out across real files and applications before returning a deliverable. A compact definition has taken hold: language models that use tools in a loop, with autonomy rising as the underlying models improve [1], [2]. This agent is no better as a conversationalist. It does the work, inside a bounded environment the person controls, and reports back. For a knowledge worker this is the decisive change. It moves the human from doing the task to supervising it, and supervising an autonomous worker is a different job from doing the task by hand.

Claude Cowork shows what the second category looks like in practice, and its name is descriptive: it is built to be a co-worker. It is an agentic desktop system, intentional and autonomous, that carries out multi-step knowledge work on a person's behalf. It operates on local files and ordinary applications and returns completed artefacts: a report, a structured spreadsheet, a folder sorted and renamed [3]. By its maker's own account it is built around the outcome, not the prompt, and built for the people who do analytical work, not for people who write software, which is to say for analysts, consultants, operations managers, and finance professionals. It runs from the desktop application without anyone writing a line of code [3]. That last point makes it the right example for this audience. The agent the MBA graduate meets first will not be a developer's instrument. It will be outcome-oriented, centred on files, and usable by someone whose training is in business, and a graduate who has never learned to direct such a thing will be unprepared for the first week of the job.

3. The gap, and why it belongs in the graded core

Most business schools do not teach what the agentic workplace demands, and the shortfall has a specific shape worth stating before turning to what should replace it.

Where AI appears in the MBA curriculum today, it tends to appear as an elective, and the elective tends to teach conversation. A course called something like "AI for business" introduces prompting, runs some basic analysis through a chat model, and shows the class how to use the tool to write, to brainstorm, to summarise a document. The treatment casts AI as an aid to thinking and a source of raw material. It rarely casts AI as something that executes a whole workflow, because until recently AI could not, and the curriculum has not caught up. Studies of how students actually use these tools bear this out. A study of postgraduate management students at a UK business school found that they used generative AI predominantly for writing, paraphrasing, and rephrasing rather than for critical analysis, with only about one in ten reaching the more demanding stages of reasoning [4]. The graduate this produces can talk to a model with some skill, and has never been taught to design a piece of work for an agent to carry out, or to supervise that agent while it does. They know how to talk to AI. Nobody taught them to direct it.

The expectation they meet at work runs ahead of this. Adoption has moved fast and clusters in exactly the knowledge-intensive, analytical tasks an MBA prepares people for, with usage concentrated where the economic value is highest [5]. Independent survey work finds that a large share of workers now use these tools and report meaningful time savings [6]. Business roles increasingly assume that an employee will hand the repetitive parts of knowledge work

to an autonomous agent, a junior co-worker who does not tire, and spend their own time on judgment and on the client. The assumption is specific. Consolidating a stack of PDFs into the skeleton of an investment memo is the agent's job. Cleaning and structuring a file of transactions is the agent's job. The first draft of a deck is the agent's job. What remains is to frame each task, checks what comes back, and owns the result. A junior hire who can only do these things by hand is slower than a peer who directs an agent, and the gap widens with every task in the week.

Here the competency stops being a nice addition and becomes something a school must grade. The ability to design and supervise a piece of agentic work is structured analytical judgment, the same faculty the MBA already claims as its central product and already examines in every core course. A programme that grades a student's reasoning in a finance case but leaves the student to absorb agentic supervision by accident is treating the second as lesser than the first, when the workplace treats them as the same kind of thing. If graduates are assumed to direct these systems, the competency has to be taught on purpose and assessed formally. It belongs in the graded core, not at the edge of the programme in an elective nobody is required to take. The framework in section 7 and the rubric it points to are what put it there.

4. Where this account comes from

This framework comes from a classroom, not from a survey of the literature. It was developed while teaching ASE-26, a course in agentic software engineering, to third-year computer science students at Holon Institute of Technology. The course ran across 3 semesters and 5 groups, with 47 students in total. It is built from 22 modules: 5 foundational lectures that establish the agentic turn and the anatomy of a workflow, 8 paired modules that each introduce a principle and then practise it, and a closing sequence on the human role, entrepreneurship, and the job market. Three projects of increasing scale run through it, and every project is submitted with an audit trail consisting of the specification, the context pack, the workflow log, and a reflection document. The audit trail is graded alongside the code. That is how the course teaches students to treat the record as part of the engineering, not as paperwork after the fact.

That teaching supplies three things: a designed and repeatedly delivered curriculum, a grading instrument applied to real student work over 3 semesters, and the accumulated experience of watching students learn to direct autonomous agents on projects large enough to fail. It supplies no controlled measurement. No comparison group was run and no instrument was validated. The failure modes named in the next section are derived from the structure of the discipline, not observed and counted in the classroom. They are analytic categories, produced by asking what can go wrong when the division of labour between a human and an agent breaks down, and they earn their place by being coherent and teachable, not merely common. A framework paper that cannot separate what it observed from what it reasoned toward is of little use to anyone who wants to test it later.

The students are not the same students, so the step to the business school needs an argument. What transfers is the structure of the relationship between a person and an autonomous agent, and that structure has nothing to do with code. A human frames the task, supplies the context,

sets the limits, and verifies the result. An agent plans and executes. A record connects the two so that the work can be defended afterwards. Every element of that sequence describes a consultant preparing a client deliverable as accurately as it describes an engineer shipping a feature. The audit trail transfers with particular force, because a business graduate is more likely than an engineer to be asked, by a manager or a client or a regulator, why a number is what it is.

Verification does not transfer, and the difference cuts against the business school. An engineering student checking an agent's work has instruments the machine cannot talk past: a compiler that refuses to build, a test suite that fails, a program that crashes on the edge case. The verification is external to both the student and the agent, and it is unforgiving. A business student has no compiler. A valuation that draws a figure from the wrong line of a statement will still compute, still balance, and still print. A competitive matrix that misclassifies a firm will still look like a competitive matrix. The check has to be performed by judgment against sources, and nothing in the artefact announces that it is needed. This asymmetry makes the competency harder to teach outside engineering and is the reason it deserves a grade. Where the environment does not enforce verification, the curriculum has to.

A second, older piece of practice stands behind the business-school half of the argument. The first author taught strategy execution and performance management at two business schools in Moscow in the mid-2000s, to students who went on to exactly the analytical and managerial roles this paper is concerned with. That experience is too old to carry evidential weight about AI, which did not exist in the classroom then. It supports something narrower: a working knowledge of what business students are like as learners, how they respond to structured analytical method, and which parts of a framework survive contact with a cohort that is impatient with theory and quick to ask what it is for. The pedagogy proposed below is written with that audience in view.

5. Translating ASE-26 pedagogy to MBA students

The framework adapts ASE-26, a discipline-based curriculum for Agentic Software Engineering [7]. ASE-26 was written for engineering students, but its central claim is not about code. It is about how a human should direct an autonomous agent and stand behind the result, and that claim holds in any field where the output is a deliverable rather than a line of software. This section extracts that claim and carries it across to business education.

5.1 A discipline, not a toolset

ASE-26 opens by treating agentic work, which we define as work delegated to an intentional, autonomous system, as a discipline. That word is chosen with care. A discipline has principles, named methods, recognised failure modes, and a standard a practitioner can be held to. It carries from one task to the next because it teaches the structure of a relationship, not the buttons of one tool. A casual user and a trained practitioner can sit at the same screen, use the same model, point at the same task, and produce different work. The gap between them is the discipline.

That reframing is what the paper borrows. Tool training ages out with each release. A discipline does not. If business schools teach Cowork as a set of features, the lesson expires when the next tool arrives. Teach the discipline of directing an agent and the lesson survives the tool. That distinction is what lets agentic AI literacy sit in the graded core instead of a short-lived elective, as section 3 argued.

5.2 The division of labour

The discipline rests on one simple split, the same split a senior takes with a capable junior. A human frames, specifies, and judges. The autonomous agent executes. It acts as a co-worker. The human's real job is to equip the agent well and to verify what it returns. Each half of that job, taken seriously, is a skill in its own right: a clear brief, the right context, sensible limits, and a careful check against a standard.

ASE-26 teaches the split by teaching the ways it breaks. The human can treat the work as their own and redo the agent's output by hand, which loses the speed and the record at once. Or they can hand over judgment along with execution and accept whatever the agent produced. Or they can hand over the framing too, let the agent decide what the problem is, and end with confident work aimed at the wrong question. Each of these is a named failure mode, and each maps onto the business analyst. The analyst who rebuilds the spreadsheet by hand, the one who ships the agent's memo unread, and the one who never asked whether the agent understood the assignment are the same three failures in a different room.

5.3 The anatomy of the work, and the audit trail

ASE-26 breaks a single turn of work with an autonomous agent into parts that recur in every domain: the intent behind the task, the brief that states it, the context the human supplies, the plan the agent forms, the execution, the verification, and the record of all of it. None of these parts is specific to software. A consultant preparing a client deck moves through the same parts as an engineer shipping a feature.

The part that matters most for business is the last one. ASE-26 makes auditability its defining principle: work that cannot be inspected, replayed, or explained after the fact is craft, while professional work is accountable and reproducible. The artefact that delivers this is the audit trail, a record that links intent to brief, to context, to what the agent did, to how the result was checked. In software this guards against silent error. For business it does more, because a manager, a client, a board, or a regulator may later ask why a number is what it is. An agent produces this trail as a byproduct of normal use, in its transcript, its sources, and its versions. The discipline asks only that the practitioner keep it, name it, and be ready to stand on it.

5.4 Verification before trust

From the division of labour follows a rule ASE-26 states plainly: every agent output is a hypothesis until it is checked. The agent works by sampling from a probability distribution, so reasoning about what it should have produced is never enough. Only a test shows what it actually produced. Empirical study of agent failures supports the rule from the other direction. A large annotated analysis of multi-agent systems found that failures of task verification form one of the three dominant categories of breakdown, alongside specification and coordination failures [8], and separate work identifies error propagation, in which an early mistake cascades through every later step, as a central bottleneck in agent reliability [9]. The discipline also names the failure that shadows the rule, verification theatre, the checking that looks rigorous and catches nothing. For MBA students this is the heart of the competency. The value they add lives in the check. Section 6 builds a deliberate planted error into the classroom for exactly this reason.

5.5 Intent that moves: the evolutionary spiral

The deepest idea ASE-26 carries is that intent itself is not fixed at the start. Drawing on Boehm's spiral model and on the co-evolution of problem and solution described by Dorst and Cross, the curriculum holds that the human rarely knows the full shape of what they want before the work begins. The first draft teaches the framer what the brief was missing. Specification and build move together across several turns, and the work converges on what was actually wanted, even though no single turn could have named the destination. Two judgments make this rhythm work: the commit point, where the practitioner decides what to lock and what to revise, and drift detection, the sense for when the work is circling or sliding off its purpose.

A business-school faculty should find this reassuring rather than alarming, because it describes how good analysts and consultants already work. The iterative pass over a deck, the memo that sharpens its own question as it is written, the model rebuilt once the first version exposes the wrong assumption: these are the spiral, already present in the case method and the capstone. No foreign habit is being imported here. It names one the school already teaches and shows how to run it with an agent in the loop.

5.6 What this means for teaching

These commitments change what is taught and what is graded. The student is no longer assessed on a finished artefact that an agent helped produce, because the agent can produce a plausible artefact on its own. Assessment falls instead on the parts the agent cannot supply: the framing, the brief, the context, the verification, and the honest record of what was delegated. ASE-26 grades the discipline, not the artefact, and that is what makes the competency both teachable and examinable. Section 7 turns these commitments into the judgment-led framework, and the assessment claim of section 3 into a working rubric.

6. The classroom, built around a mistake

Students have to do this with their hands, and a tool like Cowork is what lets them. A demonstration that goes smoothly, though, teaches the wrong lesson. If the agent takes a messy folder and returns a clean deliverable while the class watches, the students learn that the tool is impressive, which they already believed, and they learn nothing about their own role. The session has to be built so that the student's judgment is the thing on display. Build the session around a mistake.

The tasks have to be the right ones, because a demonstration only persuades if the students recognise the work as their own future. Several analytical tasks serve well, and each maps onto a course the students already take. In a corporate finance setting, the agent receives a folder of financial statements and analyst reports and is asked to pull the key metrics, build a structured spreadsheet with working formulas, and draft a first valuation summary. In a strategy setting, it takes a set of industry reports and company filings and turns them into a competitive matrix and a short executive briefing. In a consulting setting, it works from scattered interview notes, market data, and internal memos to produce the structure of a client-ready deck. For operations, it organises a body of project documentation, renames and deduplicates the files, and assembles a status report from a pile of meeting notes. One of these should carry the full session as a worked thread. The rest are better named in passing, so the class sees the range of the work without losing the single example it can follow from start to finish.

The session itself runs in three movements, and the order is deliberate. Students first take the noisy data room and structure it by hand, for ten or fifteen minutes, long enough to feel the tedium of the work and to form a private sense of what a good result would look like. They then watch the agent do the same task, faster and more completely than they could. Last, they take the agent's output and critique it, correct it, and extend it. Doing the manual pass first earns the students the standing to judge what the agent produced. A class that has not tried the task has no basis for criticism and will accept whatever the screen shows them, which is the habit the whole exercise exists to break.

The mistake is what turns the session from a demonstration into a lesson, and the case for it rests on more than pedagogy. A preregistered field experiment with consultants at Boston Consulting Group found that AI lifts performance sharply on tasks inside its capability frontier but degrades it on tasks just outside, and that the location of that frontier is not obvious to the worker even on familiar tasks, so consultants over-relied on the tool exactly where closer supervision was needed [10]. The planted error trains a student to find that boundary by hand. At least one of the agent's outputs is arranged in advance to be plausible and wrong. A valuation pulls a figure from the wrong line of a statement and carries it forward through every dependent number, so the result is internally consistent and quietly false. The competitive matrix puts a company in the wrong category, in a way that reads as reasonable until someone who knows the industry looks closely. Finding that error is the students' task, and the discovery is the point of the day. A successful run teaches them that the agent is capable. The caught error teaches them where their own value lives, and that lesson is why the competency belongs in a graded course and not in a tool tutorial. An agent is not there to be admired. It is there to be supervised, and supervision is something the student is learning to do well enough to be graded on.

ISSN 2991-9371

7. A framework built on judgment, not a checklist

A faculty needs a structure to teach against and an examiner needs one to mark against. The obvious move is a checklist of equal parts, task design, oversight, verification, ethics, one box each, and it should be resisted. A list of co-equal pillars tells a student that supervising an agent and keeping a backup are the same size of thing. They are not. The competency is one thing. It is the act of directing an intentional, autonomous system, and the other concerns are dimensions of doing that one thing well. Putting judgment at the centre keeps the framework honest about where the difficulty lies.

7.1 The spine: directing an agent is analytical judgment

At the core of the competency sits the direction of the agent, and it resolves into a handful of real sub-skills that a student can be taught and tested on. The student learns to frame a problem as an outcome rather than as a sequence of micro-instructions, which is the form an agent like Cowork is built to receive: a goal goes in, and a plan with actions comes out. She learns to structure a multi-step inquiry so the agent has a path to follow rather than a vague wish to interpret. She learns to read the intermediate outputs as they appear and to judge them in flight, instead of waiting for a final answer to accept or reject whole. And the student learns, above all, to recognise the plausible-but-wrong result, the kind the planted error in section 6 was designed to train. None of these is an exotic new ability. Each is a near relation of what the case method already drills, which is why a business school is well placed to teach the set and why the set can be examined with the same seriousness as any other analytical skill.

7.2 Oversight as a dimension of direction

An agent like Cowork is built to keep a human in the loop, so that the consequential decisions stay with the person. Inside the spine, this means a particular teachable habit: the student reviews the agent's plan before it acts, watches what it does as it works, and approves or rejects the steps that carry consequences. A simple governance checklist has a place here, as a prompt for the student to look before the agent leaps. It does not have the standing of a separate pillar. Oversight is part of directing the agent well, and it is taught and graded as such.

7.3 Verification as a dimension of direction

Verification is the sub-skill of knowing what to check and how, brought into the open and made examinable. The student cross-checks the numbers against their sources, confirms that a cited document says what the agent claims it says, tests the internal logic of the result, and pushes on the assumptions to see where they give way. Here the assessment principle from section 3 becomes concrete, because the rubric marks how the student verifies, not whether

the final answer happens to be right. Recent work on assessment design points the same way. A controlled study at a UK university found that when an AI-integrated assessment is carefully scaffolded and students are graded partly on how they engage with and reflect on the tool's output, the assessment can build both technical fluency and critical judgment [11]. A student who catches the planted error through a disciplined check has shown the competency in full, even though the artefact reached them flawed. A student who accepts a correct-looking result without testing it has not shown it, even if the result was sound. The grade follows the checking, which is the part the student controls and the part the agent cannot supply.

7.4 Ethics, privacy, and access as a dimension of direction

Giving an agent access to real files, to email, or to institutional data raises questions of data handling and privacy that the student has to learn to weigh, and these belong inside the competency, not beside it. The risks are real. Security researchers showed that a prompt injection hidden in a document could trick the desktop agent into uploading sensitive files to an attacker, and on that basis its maker advised against using it for regulated or highly sensitive data during the preview [3]. The safe operating habits are part of directing a powerful tool responsibly: work in a dedicated sandbox and not on live systems, keep sensitive personal data out of the agent's reach, review any plan that would delete or overwrite files before it runs, and take a backup before a large change. Privacy and access are part of what it means to direct the agent competently. A student who ignores them has directed it badly, which the course can say plainly and grade accordingly. One thing the framework does not settle is how much of this a school can teach before the tools change underneath it. The discipline is meant to outlive any particular agent, but that claim is a bet, and the first cohorts to be graded on it will be the ones who find out whether it was a good one.

8. Two ways to put it in the programme, and the harder problem behind both

A school persuaded that this competency belongs in the core still has to decide where it sits, and the choice is open. Better to lay out the trade-off than to prescribe an answer. The competency can live in a course of its own, a required core module on directing autonomous systems for business work, where students get sustained hands-on practice with a tool like Cowork. Or it can be woven through the existing courses, so that the finance course teaches it on a valuation, the strategy course on a competitive analysis, the operations course on a project archive, and each discipline shows the student how an autonomous co-worker behaves in that discipline's particular work. The standalone module buys depth and a clear place to put the grade, at the cost of looking like a bolt-on that the rest of the curriculum can ignore. Infusion buys integration and the lesson that agentic work is part of every function, at the cost of consistency, since it depends on a dozen instructors teaching to a common standard. A school should choose by its own structure and culture, not by a rule from outside.

The harder problem sits behind both options. Many faculty have not used these tools themselves, and a faculty member who has never directed an agent cannot teach a student to do it, still less grade the doing. Some are wary of handing a system access to their files, and that wariness is a sound instinct the course will have to address and not dismiss. The way through is to start where the risk is low. A first exposure can run on synthetic or public data, inside a folder scoped tightly enough that nothing important is within reach, on a task small enough to see all the way through. The gentle start exists to let a cautious instructor build the experience to teach from. No competency can be handed to students by a faculty that has not first acquired it, and that acquisition is the real precondition for everything this paper recommends.

9. What the graduate becomes, and what the school risks

The change lands, in the end, on the graduate, and it changes what the graduate is for. Work that used to fill the first years of an analytical career, the consolidating and cleaning and first-drafting, is the work an agent now does. What remains for the person is the part the agent cannot reach: to decide what the problem actually is, to frame the task so the agent can act on it, to bring together what several functions know, and to make the judgment that sits on top of the artefact the agent produced. This is the reconfiguration the field experiment on knowledge work anticipated, with human labour moving toward supervisory and curatorial roles rather than disappearing [10]. The graduate moves from executing the steps to directing the work and owning the decision, and the comparative advantage of an MBA, if the degree still has one, lies in that higher work. A degree has to be about the judgment now, because the labour is gone.

For the school, the consequence cuts both ways, and the downside is sharper than the upside is comfortable. The wider economy shows how early the moment is: most firms are investing heavily, yet by one large survey only a tiny fraction of leaders judge their organisations mature in their use of AI, which means the firms hiring these graduates are still working out how to deploy the technology and need people who can help them do it [4]. A programme that leaves agentic AI out, or files it under an elective, sends its graduates into those firms unable to supervise these systems safely or to get the full value from them, and the cost shows up twice, once in the employability of the graduate and once in the reputation of the school that produced them. A programme that builds the competency into the core can say something its competitors cannot, that its graduates arrive ready to direct the tools the work now requires, and that claim is worth something to students choosing a programme and to firms choosing where to recruit. Agentic AI is not one more tool to slot into an existing syllabus. It is the arrival of a new kind of co-worker, intentional and autonomous, in the room where entry-level and mid-level knowledge work happens. A change of that size has to be met at the level of the curriculum, not at the level of a single optional course.

10. Conclusion

One claim carries the paper, and a business school can act on it. Teaching MBA students to hold a skilled conversation with a generative model is no longer enough, because the work they are headed into is being reshaped by intentional, autonomous agents, new co-workers, that do the work instead of answering questions. The skill that work demands is the direction of those agents under a human's judgment. This paper contributes a diagnosis of the gap between how AI is taught and how it will be used, a framework that treats agentic AI literacy as a single judgment-led competency, and a set of teaching patterns, the planted error chief among them, that make the competency something a classroom can practise and an examiner can grade.

This paper cannot yet offer evidence that the framework works. Its claims are arguments from the shape of the technology and from one author's experience teaching the adjacent discipline to engineers, not findings from a controlled study of MBA cohorts. Only empirical work would settle the matter: measuring what students who learn this way can do that their peers cannot, then following graduates into the workplace to see whether the competency holds up against the job, then building shared teaching materials across schools so that the approach can be compared instead of asserted. Until that work is done, the case rests on reasoning. The reasoning seems to us strong enough to act on now. Waiting for evidence means graduates keep walking into a workplace the curriculum did not prepare them for, which is a more expensive way to learn the same lesson. We should also say what would embarrass this argument. If agents improve to the point where they catch their own errors reliably, the planted-error lesson becomes a period piece and the competency we describe shrinks to something a short workshop could cover. We do not think that is close, because the verification problem is not a defect of current models but a consequence of how they work. If we are wrong about that, we would rather have taught a cohort to check carefully and turned out not to need it than to have skipped the lesson and been right for the wrong reason.

References

[1] Anthropic. Building Effective AI Agents. Anthropic, 2025. https://www.anthropic.com/research/building-effective-agents

[2] Anthropic. Effective Context Engineering for AI Agents. Anthropic, 2025. https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

[3] Anthropic. Claude Cowork: Agentic AI for Knowledge Work. Anthropic, 2026. https://www.anthropic.com/product/claude-cowork

[4] Fischer, I., et al. Making Sense of Generative AI for Assessments: Contrasting Student Claims and Assessor Evaluations. The International Journal of Management Education, 2025.

[5] Anthropic. Anthropic Economic Index: Uneven Geographic and Enterprise AI Adoption (September 2025 Report). Anthropic, 2025. https://www.anthropic.com/research/anthropic-economic-index-september-2025-report

[6] McKinsey & Company. Superagency in the Workplace: Empowering People to Unlock AI's Full Potential at Work (AI in the Workplace: A Report for 2025). McKinsey & Company, 2025.

[7] Gorsky, M. ASE-26: a curriculum for agentic software engineering as a discipline. arXiv:2606.01152 [cs.CY], https://doi.org/10.48550/arXiv.2606.01152

[8] Cemri, M., Pan, M. Z., Yang, S., Agrawal, L. A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J. E., and Stoica, I. Why Do Multi-Agent LLM Systems Fail? arXiv preprint arXiv:2503.13657, 2025.

[9] Zhang, G., et al. Where LLM Agents Fail and How They Can Learn from Failures. arXiv preprint arXiv:2509.25370, 2025.

[10] Dell'Acqua, F., McFowland III, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., and Lakhani, K. R. Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality. Organization Science, 37(2): 403–423, 2025.

[11] Martin, A. F., Tubaltseva, S., Harrison, A., and Rubin, G. J. Participatory Co-Design and Evaluation of a Novel Approach to Generative AI-Integrated Coursework Assessment in Higher Education. Behavioral Sciences, 15(6): 808, 2025.

[12] Federal Reserve Bank of St. Louis. The State of Generative AI Adoption in 2025. On the Economy, Federal Reserve Bank of St. Louis, 2025. https://www.stlouisfed.org/on-the-economy/2025/nov/state-generative-ai-adoption-2025