WHITE PAPER

The ACE Model

Why Technical Excellence Alone Does Not Transform Teams

Damir Majer
& Prof. Dr. Christian Drumm
ASE Academy, Munich

May 2026
ASE ACADEMY

Contents

1 The Training Paradox 3

2 Three Dimensions, Not One 4

3 The Philosophical Foundation 5

4 What Each Dimension Looks Like 6

5 ACE and CDD 7

6 Evidence and Honest Limits 8

7 ACE in the Age of AI 9

8 The Diagnostic 10

9 From One Dimension to Three 11

Who This Paper Is For

Engineering managers, CTOs, team leads, and HR professionals who invest in developer training but still face recurring team dysfunction. This paper presents a three-dimensional model for developing complete software engineers, not just technically skilled ones.

Abstract

Technical training alone often has diminishing returns. Teams may increase analytical capability while recurring design and collaboration problems persist. The ACE Model proposes three developmental dimensions for effective software engineers: Analytical, Creative, and Emotional. The model treats these dimensions as mutually enabling rather than simply additive: severe weakness in one can limit the practical impact of the others. ACE draws on positive psychology and aligns, at a philosophical level, with the Stoic virtues. The current evidence base is observational rather than experimental. Across coaching engagements, teams that invest beyond technical training appear to develop differently from teams that do not, but causal claims remain unproven.

Foreword

This series marks twenty years of my own firm, Majer Consulting (majcon). It is not a survey of the field but the set of ideas that two decades of practice and training have most shaped, set down plainly. Read them as a practitioner's case, offered in good faith and open to challenge.

Key Takeaways

- Technical training alone often has diminishing returns. Teams that invest only in the Analytical dimension frequently improve technical knowledge without resolving recurring design or collaboration problems.
- Three dimensions, not one. The Analytical, Creative, and Emotional dimensions each shape engineering effectiveness in a different way.
- The dimensions interact. Severe weakness in one dimension can limit the practical impact of strengths in the others.
- Grounded in philosophy, not management fads. The ACE Model maps to the Stoic virtues [4] (Wisdom, Courage, Justice, and Temperance) and draws on positive psychology research.
- Start with one question. In which dimension is your team weakest? The diagnostic in Section 8 turns the model into a practical assessment tool.

1. The Training Paradox

Every organization I work with tells the same story. They send developers to conferences. They buy online courses. They run internal tech talks. They invest in certifications. And still, the same categories of team problems come back. Not the same problems. The same categories.

Strong technical skills. Still ships over-engineered solutions. Clean architecture knowledge. Still cannot simplify a design when the deadline approaches. Advanced framework expertise. Still writes code reviews that demoralize junior developers.

The industry treats developer growth as a one-dimensional problem: more technical knowledge equals better developer. Weinberg [17] identified this blind spot in 1971: programming is a human activity first and a technical one second. Brooks [18] showed that communication overhead grows faster than team size, dominating the cost of scaling a project.

Growth = Development Across Three Dimensions

We measure developer capability by what they know. The better question: what can they do with what they know, and can they do it with other people? Technical skills are necessary. They are not sufficient.

2. Three Dimensions, Not One

> The critical factor is the human in all cognitive directions.
>
> Adapted from positive psychology

The ACE Model proposes three distinct dimensions that together define the complete software engineer.

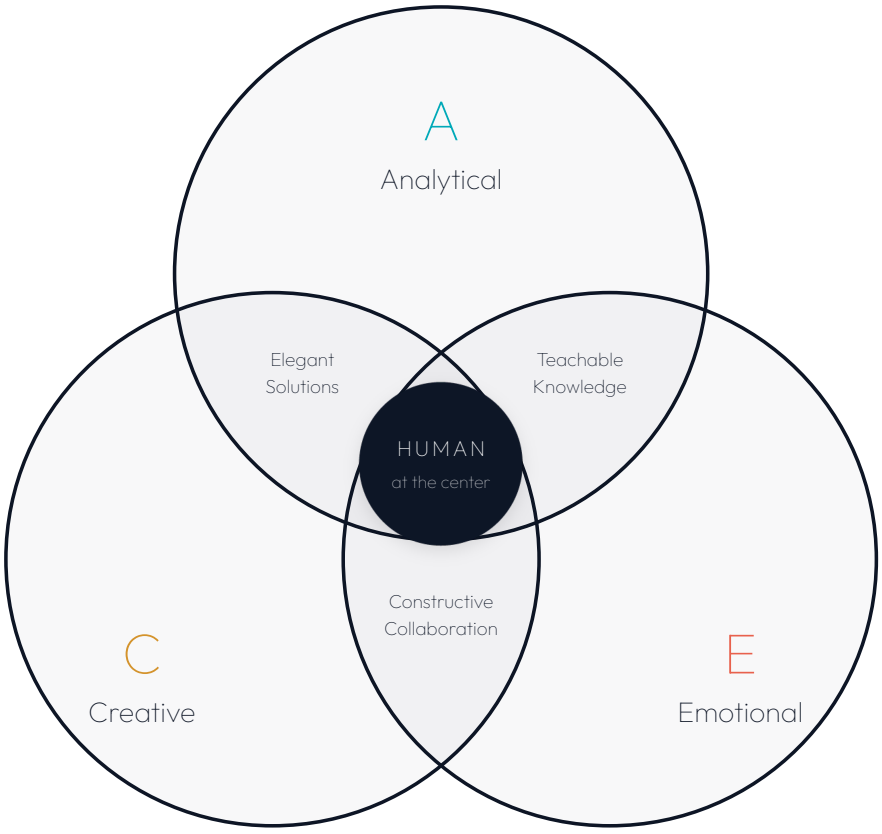

Fig. 1. The ACE Model. Three dimensions with the human at the center; the overlapping areas show the compound capabilities they produce.

A – Analytical

Systems thinking. Developer mindset. Domain knowledge. State-of-the-art technology. Learning-to-learn. This is where the vast majority of training budgets go.

C – Creative

Design thinking. Associated thinking. Simplicity focus. Solution finding. The dimension that separates "it works" from "it is elegant."

E – Emotional

Collaboration. Understanding others. Finding compromises. Valuing yourself and the team. Ethics. The dimension that determines whether good ideas land.

The three dimensions are not presented as independent contributions that simply add up. In practice, they act as mutually enabling conditions. A severe weakness in one dimension can sharply reduce the practical effect of strength in the others.

Developer Impact $A \times C \times E$

The formula is heuristic rather than literal. It represents interdependence: a developer strong in the Analytical and Creative dimensions but severely weak in the Emotional dimension may still produce work that fails to land in a team setting.

The Compound Capabilities

The overlapping areas in Fig. 1 are where the real leverage lives:

- Analytical × Creative → Elegant Solutions. Technical skill combined with creative thinking produces designs that are not just correct but simple.
- Analytical × Emotional → Teachable Knowledge. Technical depth combined with empathy produces developers who can explain and mentor effectively.
- Creative × Emotional → Constructive Collaboration. Creative problem-solving combined with emotional awareness produces teams that disagree productively.
- Analytical × Creative × Emotional → The Complete Engineer. Elegant code, clear explanations, constructive feedback, simple solutions to complex problems.

3. The Philosophical Foundation

The model's structure raises a question: why these three dimensions and not others? The answer does not come from management theory. It comes from positive psychology and a much older tradition.

The original insight came during coursework at the Inntal Institute [10]: the critical factor in any cognitive endeavor is the human. Not the tools. Not the methodology. Not the framework. The human, in all cognitive directions.

This led to a question: what are the cognitive directions that matter for software engineering? Not just the technical direction. All directions. The idea of multi-dimensional development has historical precedent: Pestalozzi's "Head, Heart, Hand" [3] argued for intellectual, emotional, and practical education as early as 1801. The three ACE dimensions are interacting peers: none sits above the others. (This is a different axis from Bloom's taxonomy, which describes increasing depth of understanding and can be applied within any one of the three dimensions, rather than ranking them against each other.)

The Stoic Connection

The mapping to the Stoic virtues was discovered after the model existed, not designed into it. But the correspondence is striking:

STOIC VIRTUE	ACE DIMENSION	WHAT IT MEANS FOR ENGINEERS
Wisdom	Analytical	Understanding systems deeply. Knowing when to apply which principle. Learning from experience.
Courage	Creative	Daring to simplify. Proposing unconventional solutions. Challenging "we have always done it this way."
Justice + Temperance	Emotional	Treating colleagues fairly. Giving honest feedback with empathy. Knowing your limits. Valuing the team above your ego.

The Stoics argued that these virtues are not separate achievements but a unified whole. You cannot have true wisdom without the courage to act on it, or justice without the temperance to apply it wisely. The same holds for ACE: the dimensions are inseparable in practice.

The Shape of a Complete Engineer

The model was first sketched in 2018 as the ICE Model (IT, Creative, Emotional) and renamed to ACE in 2026 through collaborative sessions with Prof. Dr. Christian Drumm [2]. The visual representation evolved from Double-T-Shaped to Pi-Shaped to M-Shaped [1], grounded in positive psychology and complex learning theory. The result: not a specialist who can also communicate, but a professional whose analytical, creative, and emotional capabilities are each developed to depth.

---

4. What Each Dimension Looks Like

Abstract models need concrete examples. Here is what each ACE dimension looks like in daily engineering work, drawn from coaching observations across teams.

The Analytical Dimension in Practice

In the AI era, the Analytical dimension expands to include the formalization of intent: the bridge between a creative vision and a technical constraint. A high-ACE engineer does not just write code; they turn messy human requirements into precise, machine-actionable specifications. Architecting through intent becomes the primary differentiator when raw coding execution is commoditized.

A beginning developer knows that "functions should be small." An experienced developer sees the Integration/Operation Segregation Principle (IOSP) [5]: every function is either an integration or an operation, never both. Same principle. Deeper perception. Different category of bugs eliminated.

The Creative Dimension in Practice

A developer weak in the Creative dimension codes the first solution that comes to mind. A developer strong in it generates three approaches, evaluates trade-offs, and selects the simplest. The difference is the quality of the design space explored.

The Emotional Dimension in Practice

Consider code reviews. A developer strong in the Analytical dimension but weak in the Emotional one writes: "This is wrong. Use the strategy pattern." A developer strong in both writes: "I see what you were going for. Here is a structural alternative that avoids the conditional chain. Want to pair on it?" Same technical content. Completely different outcome. The first creates defensiveness. The second creates learning.

DIMENSION	OBSERVABLE METRIC	WHAT "HIGH" LOOKS LIKE
Analytical	Concept transfer to new contexts	Applies IOSP in a codebase they have never seen
Creative	Alternative solutions per problem	Generates 3+ approaches before selecting one
Emotional	Empathy in code review	Reviews that make colleagues better, not defensive

Scoring Heuristic

Coaches and team leads need a shared language for these dimensions before they can develop them. To that end we propose a provisional rubric scored from 1 to 5, used in our coaching engagements and equally available to a team that wants to locate its own weakest dimension. The rubric is intended for structured discussion, comparative assessment, and intervention design. It is not a validated psychometric instrument, and its scores should be interpreted ordinally and directionally.

SCORE	ANALYTICAL (A)	CREATIVE (C)	EMOTIONAL (E)
1	High coupling, global state, "big ball of mud" logic.	Single solution only; no trade-off analysis.	Personal judgment in reviews; creates silos.
2	Follows rules but misses deeper patterns (e.g., IOSP).	Considers two options but defaults to "first to mind."	Technically correct feedback but cold or overwhelming.
3	Consistent use of basic constraints (Value Objects, etc.).	Evaluates 3+ approaches; selects for simplicity.	Constructive language; focuses on code, not person.
4	Transfers concepts across contexts; teaches others.	Proactively simplifies; challenges "over-engineering."	Focuses on author's learning; reviews empower.
5	Structural quality is a property of the work, not an effort.	Design space exploration is the primary activity.	Reviews are sought out as growth; builds safety.

Usage: Two independent raters score each team member. Disagreements of more than 1 point on any dimension trigger a calibration discussion. The combined profile is intended to identify intervention priorities, not to produce a precise numeric ranking. A team strong in the Analytical and Creative dimensions but weak in the Emotional one, for example, has a clear developmental target, even though the overall score is only a rough heuristic.

5. ACE and CDD

The companion white paper, Constraint-Driven Development (CDD) [9], introduces a six-level Constraint Stack for reducing recurring defect classes through structural constraints. CDD addresses the Analytical dimension of ACE. It is rigorous, systematic, and effective. And it is one-third of the picture.

Constraints require all three dimensions to work in practice:

- Analytical: Understanding which constraint to apply at which level. This is the technical knowledge that CDD provides.
- Creative: Finding elegant implementations of constraints. A NonEmptyList type is a constraint. An elegant NonEmptyList type that integrates naturally into the domain model draws on the Creative dimension.
- Emotional: Making constraints stick in a team. When a junior developer violates a constraint, does the team respond with judgment or with coaching? The emotional dimension determines whether structural improvements are sustained or abandoned.

Fig. 2. The ACE Board. A 3×3 matrix measuring development across all three dimensions. Rows: A, C, E. Columns: what they produce (Artifacts), what they know (Knowledge), and how they behave (Behaviour).

The ACE Board draws together the assessment tools ASE Academy already uses, each of which covers part of the picture. The Code Quality Notebook reviews the artifacts a developer produces, which speaks to the Analytical dimension. The CRAFT Calisthenics, ASE Academy's set of principles for writing changeable software (the practitioner's counterpart to the better-known Object Calisthenics), build knowledge across the Analytical and Creative dimensions. Key Behavior Indicators capture observable habits in review and collaboration, which touch all three dimensions. The Board combines them into one view so that no dimension is assessed in isolation.

6. Evidence and Honest Limits

What We Know

Psychological safety is the top predictor of team effectiveness. Google's Project Aristotle [15], studying 180 teams, found psychological safety — not technical skill — was the strongest predictor of team performance. Edmondson [16] formalized this as the foundation of high-performing teams. Both findings support the E dimension as a structural enabler, not a "soft" supplement.

Emotional intelligence predicts team performance. Goleman [6] found that emotional competencies distinguished top performers from average ones at roughly twice the rate of cognitive ability alone, across 181 job roles.

Positive psychology provides the theoretical foundation. Seligman [7] on character strengths supports the premise that human flourishing requires development across multiple dimensions, not just cognitive skill.

The employee advantage is real. Meier [8] at Columbia Business School demonstrates that organizations investing in employee development outperform those treating people as interchangeable resources.

Coaching observation. A development team at a European insurance company (15 developers, Java and web stacks, 7-year coaching engagement) showed a substantial reduction in production bugs over a multi-year period that included interventions targeting Creative development (design simplification, alternative generation before coding) and Emotional development (code review workshops emphasizing empathy, pair programming rotations), in addition to already-strong technical training. Because the intervention was uncontrolled and co-occurred with other organizational changes, the case should be read as an illustrative practice observation rather than causal proof.

What We Do Not Know

No randomized controlled trial exists; the evidence base is observational. The multiplicative formula is a heuristic representation of interdependence, not a mathematical law. The scoring rubric is useful for practice but has not been psychometrically validated. A strong counterexample would be a team with persistently weak Emotional effectiveness that nonetheless sustains high performance over time, or robust evidence that technical training alone resolves recurring collaboration dysfunction.

7. ACE in the Age of AI

These limits acknowledged, the model gains relevance in a new context. AI tools may commoditize parts of analytical execution: code generation, pattern recall, and some forms of review assistance. One controlled study illustrates the shift: METR [11] (a single small-sample RCT, n=16 experienced open-source developers) found that participants using early-2025 AI tools were actually 19% slower, which METR attributed to verification overhead and task-switching cost. A study of this size is suggestive, not conclusive. A larger, peer-reviewed 2026 study points the same way from the code itself: across 5,000 AI-refactored programs, CodeScene [21] found that defect risk rose sharply in less-healthy code, concluding that code quality is now a prerequisite for the safe and effective use of AI. We read both signals through the ACE framing: when generated output becomes easier to produce, judgment, design choice, and social effectiveness become more visible constraints. These connections to the ACE Model are ours, not the original authors'.

When analytical execution becomes easier to access, the Creative and Emotional dimensions can become stronger differentiators of impact.

> Your ability to design systems, empathize with users, and make nuanced judgment calls will only become more valuable, not less.
>
> Addy Osmani, Google [12]

Kent Beck [13] frames it precisely: "In a world of abundant cheap code, what becomes scarce is understanding, judgment, the ability to see how pieces fit together." Understanding is Analytical. Judgment is Creative. Seeing how pieces fit together for real people is Emotional.

The shift from "human-in-the-loop" to "humans-above-the-loop" [14] means providing judgment, creativity, and empathy while AI handles execution. This is the ACE Model applied to the AI era. The human remains at the center. The tools change. The dimensions do not.

McKinsey's 2026 hiring shift provides a suggestive parallel: the firm is reportedly prioritizing liberal arts majors over pure analytics hires, because creativity and judgment matter more than raw analytical skill when AI agents handle analytical work [19]. This is not proof of the ACE Model, but it is directionally consistent with it.

The heuristic still applies: when one dimension becomes easier to outsource or augment, the remaining dimensions matter more, not less. None of these parallels constitutes proof. Multiple directionally consistent signals do not add up to a causal demonstration.

8. The Diagnostic

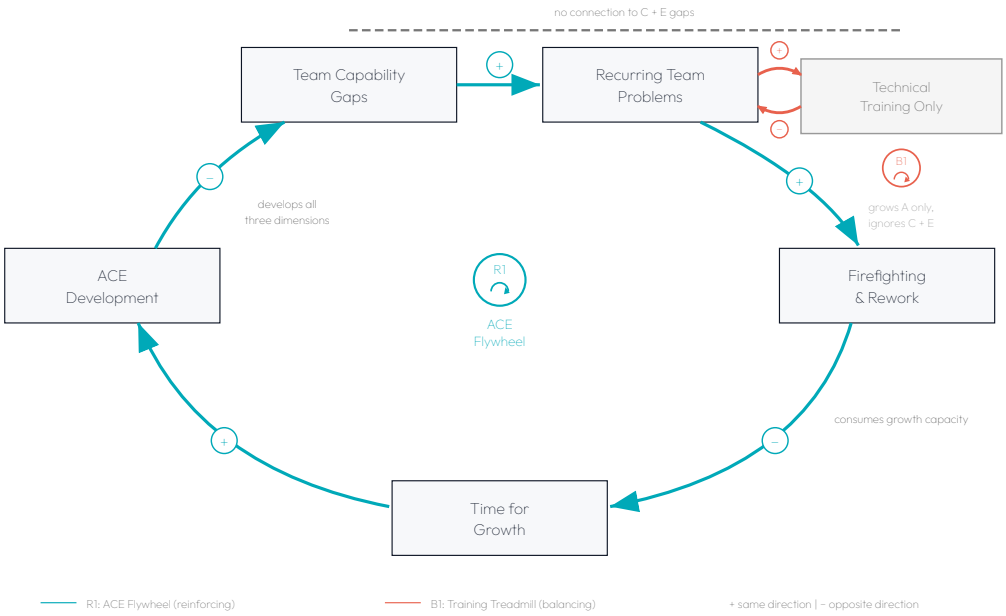

Fig. 3. R1 (teal): the ACE Flywheel. Multi-dimensional development reduces capability gaps, freeing time for growth. B1 (coral): the Training Treadmill. Technical training grows only A, leaving C + E gaps unaddressed.

Figure 3 traces two feedback loops. The reinforcing loop R1, which we call the ACE Flywheel, is the engine of compounding growth: developing all three dimensions narrows team capability gaps, which means fewer recurring problems, less firefighting, and more time for the next round of growth, so the loop turns again. The balancing loop B1, the Training Treadmill, is why technical-only investment stalls: it grows the Analytical dimension alone, so the Creative and Emotional gaps regenerate the same problems and the team runs hard without advancing.

To apply the ACE Model as a diagnostic:

1. **Assess All Three Dimensions**
   Evaluate the Analytical, Creative, and Emotional dimensions independently. Use the observable signals from Section 4.

2. **Identify the Weakest Dimension**
   Where do recurring team problems cluster? Communication breakdowns suggest E. Over-engineered solutions suggest C. Persistent bugs suggest A.

3. **Develop One Dimension**
   Not all three at once. Weak Emotional: code review workshops focused on empathy. Weak Creative: design-thinking exercises before coding. Weak Analytical: invest in technical craft through structural constraints.

4. **Observe the Compound Effect**
   Improving one dimension unlocks the others. Better collaboration (E) creates space for creative exploration (C). Simpler designs (C) make concepts easier to teach (A). The flywheel starts.

9. From One Dimension to Three

The question is not whether your team needs better developers, but what “better” means. Most organizations define it along a single axis: technical knowledge. The same categories of team problems persist because they are capability gaps in dimensions that nobody is developing.

The ACE Flywheel shows how multi-dimensional development compounds. The Training Treadmill shows why technical-only investment plateaus.

ASE Academy develops ACE engineers. Analytical. Creative. Emotional. The human at the center. Start with one question: In which dimension is your team weakest? Not better coders. Better engineers. That is the shift.

This paper is part of a trilogy. The ACE Model addresses who develops well. CDD [9] addresses how to enforce quality. The Economics of Changeability (EoC) [20] addresses what to optimize for. Fig. 4 shows how the three connect.

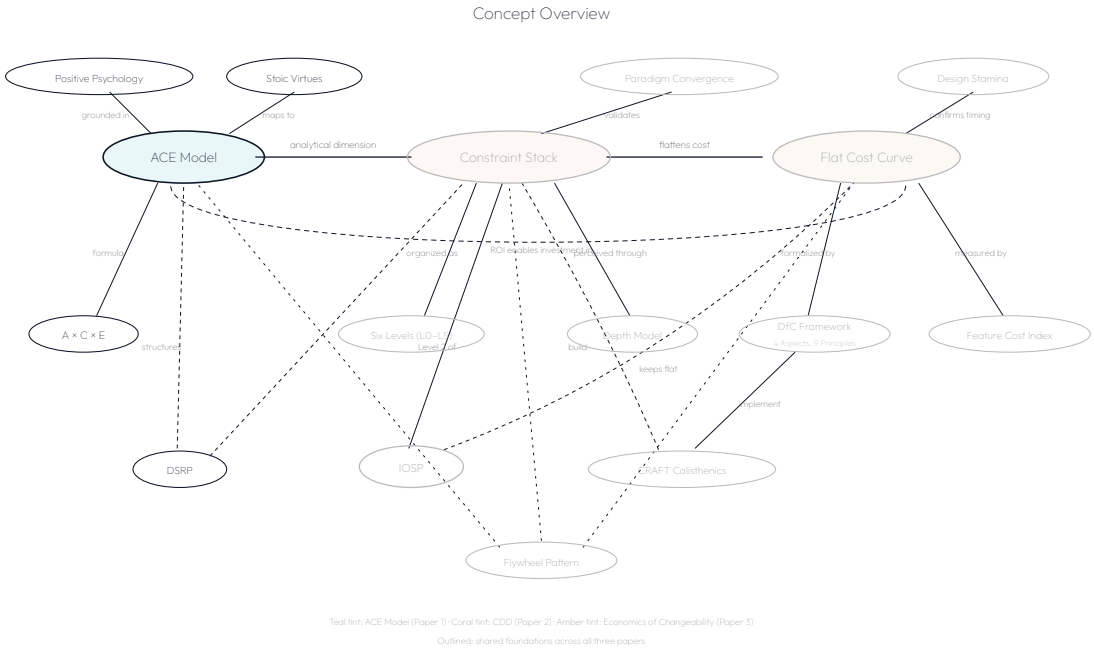

Fig. 4. Concept overview of the trilogy. Teal: ACE Model (this paper). Coral: Constraint-Driven Development. Amber: Economics of Changeability. Navy: shared foundations.

Appendix: Methods

The claims in this paper rest on coaching observations across development teams in European enterprises (insurance, automotive, logistics, SaaS), spanning approximately 15 years. No controlled experiment has been conducted. The evidence is qualitative and observational.

Instrument: ACE Scoring Heuristic

The 1–5 scoring rubric (Section 4) is designed for team diagnostics. Two independent raters assess each team member on all three dimensions. In a preliminary pilot application with senior technical leads, the rubric showed promising inter-rater agreement. This result is provisional: the sample was small, the raters were not blinded, and the instrument has not yet been formally validated.

Falsifiability

The model makes two testable claims: (1) engineering effectiveness is better described by interacting dimensions than by a single technical axis; and (2) interventions that address neglected dimensions may outperform purely technical interventions in teams where collaboration or design quality is the limiting factor. Strong counterexamples would weaken the model.

Limitations

- No randomized controlled trial. Causality is plausible but unproven.
- The multiplicative formula (A × C × E) is a heuristic model, not a mathematical law. The “units” of each dimension are ordinal, not interval.
- Sample is biased toward European enterprise teams with existing coaching relationships.
- The scoring rubric has not been validated psychometrically. It is a practitioner instrument, not a standardized assessment.

References

[1] Majer, D.: M-Shaped Skills and the Polymath Developer. Internal working paper, ASE Academy (2026)
[2] Majer, D., Drumm, C.: The ACE Model. Developed collaboratively, based on positive psychology foundations (2026)
[3] Pestalozzi, J.H.: Head, Heart, Hand. Educational philosophy (1801). Historical precedent for multi-dimensional development
[4] Marcus Aurelius, Seneca, Epictetus: Stoic virtues. Wisdom, Courage, Justice, Temperance. Various works, classical antiquity
[5] Westphal, R.: Integration Operation Segregation Principle. ralfwestphal.substack.com
[6] Goleman, D.: Emotional Intelligence. Bantam Books (1995); Working with Emotional Intelligence. Bantam (1998)
[7] Seligman, M.: Authentic Happiness. Free Press (2002); Flourish. Atria Books (2011)
[8] Meier, S.: The Employee Advantage. PublicAffairs (2024). Columbia Business School
[9] Majer, D.: Constraint-Driven Development. ASE Academy White Paper (2026)
[10] Inntal Institut: Positive Psychology Professional Training. Munich
[11] METR: Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. RCT with 16 developers (2025)
[12] Osmani, A.: Beyond Vibe Coding: From Coder to AI-Era Developer. O’Reilly (2025)
[13] Beck, K.: Programming Deflation. Tidy First? Substack (2025)
[14] Diginomica: From Human-in-the-Loop to Humans-Above-the-Loop (2026)
[15] Duhigg, C.: What Google Learned From Its Quest to Build the Perfect Team. NY Times (2016). Project Aristotle: 180 teams
[16] Edmondson, A.: The Fearless Organization. Wiley (2019)
[17] Weinberg, G.: The Psychology of Computer Programming. Dorset House (1971)
[18] Brooks, F.: The Mythical Man-Month. Addison-Wesley (1975)
[19] Roose, K.: McKinsey Starts Hiring More Liberal Arts Majors. Fortune (Feb 2026). AI agents handle analytical work; creativity and judgment become the differentiator
[20] Majer, D.: The Economics of Changeability. ASE Academy White Paper (2026)
[21] Borg, M., Hagatulah, N., Tornhill, A., Söderberg, E.: Code for Machines, Not Just Humans — Quantifying AI-Friendliness with Code Health Metrics. arXiv:2601.02200; ACM (2026). 5,000 programs; AI defect risk markedly higher in unhealthy code

About the Authors

Damir Majer
Damir Majer coaches development teams on software quality at ASE Academy in Munich. The ACE Model emerged from his Positive Psychology studies at the Inntal Institute and years of coaching teams across industries.
Contact: d.majer@majcon.de | ase.academy

Prof. Dr. Christian Drumm
Christian Drumm is a professor of Information Systems at FH Aachen and co-developer of the ACE Model. His work bridges teaching, software engineering, and hands-on coaching across industries.
Contact: cd@trans4m.io

majcon — Assessment & Diagnostics · ASE Academy — Transformation & Coaching

CRAFT Calisthenics is part of the Craftology® methodology by ASE Academy.