Fra magisk tænkning til aktiv virkelighedsprøvning

Udvikling af en taksonomisk model og didaktisk intervention mod kognitive barrierer hos novicer i programmering

Abstract

This working paper addresses a critical challenge in computing education: the severe cognitive barriers and high dropout rates experienced by novice programmers. Grounded in action research and a deductive-hypothetical approach, this study introduces an innovative theoretical bridge between IT-didactic research and Sam Vaknin's epistemological framework of psychodynamic development, with a specific focus on the incorporation stage and Reality Testing (RT). Across seven empirical investigations involving 15 students—combining broader classroom investigations of 12 students with in-depth tracking of three focal students, culminating in an intensive single-case study—the research establishes a new four-level taxonomic model of programming cognition (Use, Adjust, Destroy, Put together) mapped across two emotional zones (Safety vs. Anxiety). The findings demonstrate that introducing structured didactic interventions forces students out of superficial pattern-matching and into active Reality Testing, navigating a demanding cognitive dissonance rooted in Self-Determination Theory. Ultimately, the study highlights the vital psychological judgment required of educators to guide novices toward robust mental models and authentic mastery.

---

1. Introduktion

Inden for it-didaktikken er det en veldokumenteret og kritisk problemstilling, at ca. 30 % af nybegyndere (novicer) oplever massive kognitive vanskeligheder, når de skal tilegne sig grundlæggende programmeringsforståelse, hvilket medfører en høj risiko for frafald (Simon et al., 2019). Hvor programmeringsfærdigheder historisk blev testet i isolerede kodetests, er programmeringsevalueringen i dag typisk integreret i tværfaglige projekter sammen med systemudvikling. Denne integrerede form gør det muligt for svage studerende at kompensere for kognitive barrierer ved at levere stærke procesorienterede eller metodiske præstationer, hvilket reelt kan skjule manglende individuel programmeringsforståelse.

Formålet med dette projekt er at undersøge de bagvedliggende mentale processer og udvikle en didaktisk indsats, der understøtter reel strukturel forståelse frem for overfladisk mønstergenkendelse.

2. Baggrund og litteraturafsøgning

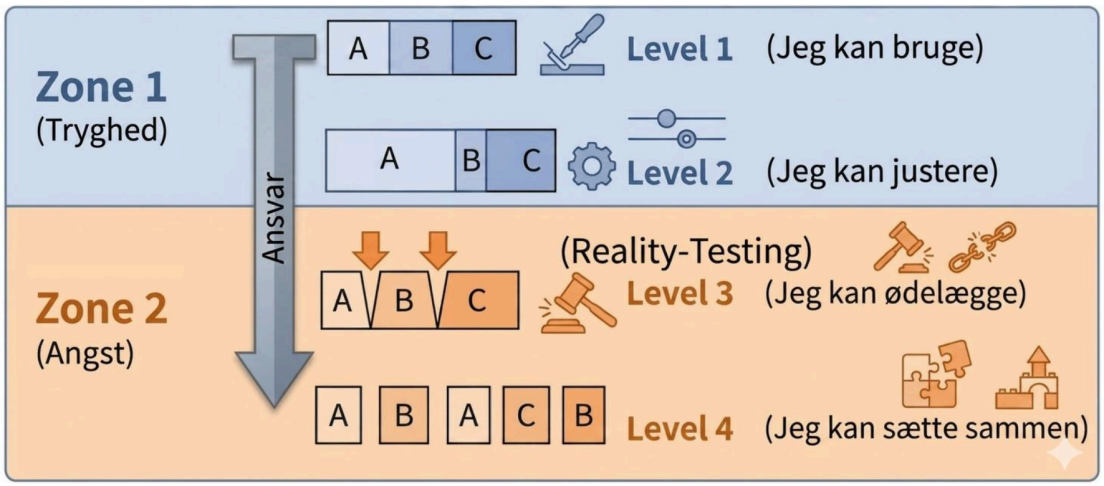
Eksisterende it-didaktisk og kognitionspsykologisk forskning konstaterer traditionelt, at novicer oplever massive barrierer og kognitiv overbelastning (Sweller, 1988), fordi de mangler funktionelle mentale modeller af computerens eksekvering – en såkaldt *Notional Machine* (Norman, 1983; Caspersen, 2007; Ma, 2007; Sorva, 2013) – samt systematiske strategier til kodelæsning og *tracing* (Lister et al., 2004; Xie et al., 2018; Enebjörk, 2022). Projektets teoretiske fundament bygger på en nyskabende brobygning til den israelske forfatter og psykolog Sam Vaknins epistemologiske model for udvikling (Vaknin, 2024). Vaknin systematiserer fire klassiske begreber fra psykoanalysen og objektrelationsteorien – *internalisering, introjektion, identifikation og inkorporering* (Freud, 1923; Klein, 1932; Laplanche & Pontalis, 1973; Olsen, 2026) – til en sekventiel fire-trins model og anvender den på videnstilegnelse. I forbindelse med *inkorporering* indtræffer *Reality Testing (RT)*; her realiserer den erkendende part adskillelsen mellem selvet og viden, hvilket gør det muligt at teste og validere informationens sandhedsværdi frem for blot at fusionere med den.

Hypotesen er, at de mangelfulde mentale modeller kognitivt kan forstås som et svigt i den studerendes virkelighedsprøvning, hvor en "indre fantasi" eller magisk tænkning overstyrer computerens objektive feedback.

3. Teoretisk ramme og den taksonomiske zonemodel

Projektet bygger på en brobygning mellem it-didaktik og den israelske psykolog Sam Vaknins epistemologiske model (internalisering, introjektion, identifikation og inkorporering). På baggrund af projektets empiriske data er der foretaget en metodisk justering og didaktisk operationalisering, hvor der i princippet er byttet rundt på Vaknins trin 3 og 4: Hvor Vaknin placerer Reality Testing og inkorporering som det afsluttende fjerde stadie, viser dataene her, at evnen til selvstændigt at 'sætte sammen' (mestring på Level 4) forudsætter en forudgående, aktiv Reality Testing (RT). Dermed rykkes RT frem som den aktive dekonstruktionsmetode allerede på Level 3 ('ødelægge'), hvilket muliggør den fulde integration og rekonfiguration på Level 4.

På baggrund heraf er der udviklet en visuel zonemodel bestående af to overordnede zoner og fire niveauer, som illustreret i figuren nedenfor:



Figur 1: Den taksonomiske zonemodel for kognitiv progression hos novicer, der illustrerer bevægelsen fra skabelon-tryghed til aktiv Reality Testing og inkorporering (Egen tilvirkning, Hansen, 2026a).

Zonemodellen opdeles i:

- **Zone 1 (Tryghed) - Skabelon-området:**
  - *Level 1 (Brug / Internalisering):* Koden opfattes som en lukket blok og et fremmed objekt; computerens feedback ignoreres til fordel for "magisk tænkning".
  - *Level 2 (Juster / Introjektion):* Skabelonen er indlejret, så overfladiske parameterjusteringer er mulige, men den studerende er fortsat bundet af den visuelle skabelon.
- **Zone 2 (Angst) - Model-området:**
  - *Level 3 (Ødelæg / Inkorporering & RT):* Dekonstruktionsfasen, hvor didaktiske interventioner skaber kognitiv dissonans, bryder den trygge symbiose med skabelonen op i isolerede komponenter og tvinger den studerende til at adskille egne antagelser fra computerens objektive virkelighed gennem aktiv Reality Testing.
  - *Level 4 (Sæt sammen / Identifikation & Rekonfiguration):* Når virkelighedsprøvningen er gennemført, kan den studerende udholde frustrationen, gøre kodens logik til sin egen og selvstændigt rekonfigurere brikkerne i en ny, stabil mental model kalibreret mod computerens objektive virkelighed, hvor RT er blevet en iboende kompetence.

Pædagogisk illustration: At huske ruten vs. at kende slottet

Forskellen på Zone 1 og Zone 2 – og den mentale tilstand, den studerende befinder sig i – kan illustreres gennem en rumlig metafor: Forestil dig to personer, der bor på et stort slot i en uge.

Et år senere står de i hovedindgangen og bliver bedt om at finde hen til *"værelset med den lille bruser"*.

- **Person A i Zone 1 (Falsk tryghed og passivitet):** Under hele opholdet har Person A befundet sig i en bekvem, uforpligtende tilstand som passiv passager. Vedkommende har blot fulgt efter andre eller klynget sig til en isoleret opskrift som *"drej til venstre efter dør 2"*. Fordi Person A aldrig har taget ansvar for at undersøge, hvordan slottets fløje reelt hænger sammen, har hjernen kun lagret overfladiske pejlemærker uden strukturel forståelse (Level 1–2). Et år senere finder Person A derfor ikke frem til værelset med den lille bruser; så snart en dør er låst, eller udgangspunktet ændres, er vedkommende helt fortabt – trygheden viser sig at være en stagnationsfælde.
- **Person B i Zone 2 (Nysgerrighed, kognitiv dissonans og aktiv virkelighedsprøvning):** Person B har derimod valgt – eller er via didaktiske vejbump blevet skubbet til – aktivt at udforske slottet på egen hånd. I en tilstand af nysgerrighed og produktiv usikkerhed ("angst-zonen") har vedkommende åbnet ukendte døre, gået ad bagtrapper og faret vild med vilje for at teste, hvordan rummene hænger sammen i virkeligheden (Level 3: *Ødelæg / Reality Testing*). Ved at udholde denne frustration og konfrontere bygningens faktiske indretning i stedet for at gætte, har Person B inkorporeret et komplet mentalt kort over slottet (Level 4: *Sæt sammen*) og finder et år senere ubesværet frem til værelset med den lille bruser fra et hvilket som helst udgangspunkt.

4. Metodetilgang

Undersøgelsen er designet som et didaktisk eksperiment, der integrerer principper fra aktionsforskning (Hastrup et al., 2026) med en hypotetisk-deduktiv tilgang (Nielsen, 2026). Aktionsforskningen muliggør en iterativ proces, hvor didaktiske interventioner løbende udvikles og justeres i tæt samspil med undervisningspraksis. Den hypotetisk-deduktive tilgang fungerer som det videnskabelige fundament, hvor Vaknins model udgør den testede hypotese.

Datagrundlaget er indsamlet fortløbende fra august 2025 til februar 2026 i egen undervisningspraksis på datamatikeruddannelsen og omfatter i alt 15 studerende fordelt på syv kronologiske undersøgelser: 12 studerende på 2. semester, der indgår i de bredere, tværgående klasseundersøgelser, samt 3 studerende på 3. semester, der følges dybdegående, herunder ét særligt intensivt casestudie.

4.1. De iterative cyklusser og didaktiske handlinger

Selve aktionsforskningen blev struktureret omkring tre overordnede iterative cyklusser, der fungerede som den didaktiske motor for undersøgelserne:

- **Iteration 1 (Den diagnostiske afdækning):** Fokuserede på observation af de studerendes naturlige adfærd uden didaktisk indgriben for at afdække omfanget af de kognitive barrierer (Undersøgelse 1 og 4).
- **Iteration 2 (Kognitive knudepunkter og interventioner):** Fokuserede på at teste og justere konkrete pædagogiske værktøjer. Handlingen bestod først i at redesigne visualiseringsværktøjet CodeToModel (CtM) (Hansen, 2026c), så variabler blev ændret fra statiske kasser til dynamiske "pegepinde", der visuelt adskiller deklaration fra tildeling (Undersøgelse 2). Derudover blev der grebet ind i arbejdstempoet ved at introducere test-cases i udviklingsmiljøet CodeExercises (CE) (Hansen, 2026b) (Undersøgelse 3). Endelig blev der indført didaktiske "hastigheds-vejbump" i form af tvungne skriftlige forudsigelser (predictions) før kørsel af kode, og undervisningsformen blev ændret fra asynkrone lister til synkrone, fælles opgaver (Undersøgelse 5).
- **Iteration 3 (Validering og systemisk ændring):** Fokuserede på at teste effekten af interventionerne. En konkret handling var et kontrolleret klasseeksperiment, som medførte en permanent omlægning af det digitale Quiz System (QS) (Hansen, 2026d) fra multiple-choice til krav om manuelle, numeriske input for at gennemtvinge mental kode-eksekvering (Undersøgelse 6). Til sidst blev interventionernes langsigtede effekt testet i et dybdegående casestudie, valideret af en 100 % objektiv kodetest (Undersøgelse 7).

(Se Bilag 1 for en uddybende beskrivelse af formål og testmetoder i de enkelte undersøgelser).

5. Resultater (Indsigter fra undersøgelserne)

Undersøgelsernes tre faser leverede følgende primære indsigter:

- **Indsigter fra Iteration 1:** Det blev dokumenteret, at novicer (bl.a. Person A og B) i høj grad opererer med "magisk tænkning", hvor indre forventninger overstyrer computerens objektive skærmfeedback. Konklusionen var et totalt fravær af virkelighedsprøvning; de studerende gættede og ignorerede den faktiske kode-virkelighed.
- **Indsigter fra Iteration 2:** De kvantitative tests i Quiz System (QS) (Hansen, 2026d) påviste massive forståelseshuller på det fundamentale niveau, særligt i opgaver hvor den studerende skulle spore sekventielle værdiændringer på tværs af flere variabler. Det blev tillige afsløret, at test-tænkning slet ikke var en del af novicernes bevidsthed. Interventionerne viste imidlertid, at når studerende tvinges ned i kognitivt tempo via skriftlige forudsigelser i CodeExercises (CE) (Hansen, 2026b), afskæres de fra deres vante mønster med at rette i blinde og tvinges i stedet til at konfrontere deres fejlbehæftede mentale modeller.
- **Indsigter fra Iteration 3:** Klasseeksperimentet uden computere demonstrerede, at stærke studerende spontant anvendte konkrete tal til at "hoved-debugge" logikken,

mens svage studerende strandede på overfladisk syntaks, hvilket bekræftede nødvendigheden af at fjerne multiple-choice for at stoppe gætteri. Det dybdegående casestudie af Person C udgjorde undersøgelsens største gennembrud: Det beviste, at de kontinuerlige Reality Testing-interventioner kan skabe en dybdegående kognitiv forandring, hvor en tidligere skabelon-fastlåst studerende formår at dekonstruere avanceret kode selvstændigt og aktivt fravælge AI-værktøjer for at bygge ægte mestring.

6. Diskussion

Analysen viser, at der er en direkte rød tråd mellem den teoretiske forståelse og det, der sker i klasselokalet. Når vi overfører psykoanalytiske principper om Reality Testing til it-didaktikken, opdager vi, hvorfor de studerende falder i "falsk tryghed"-fælden (Zone 1) med overfladiske skabeloner og ukritisk AI-brug.

De målrettede didaktiske interventioner fungerer som det nødvendige spejl: De synliggør for den studerende selv, at deres indre antagelser ikke stemmer overens med computerens virkelighed. Denne erkendelse skaber en sund, men ofte smertefuld kognitiv dissonans (Zone 2). For de studerende er denne proces forbundet med frustration og modstand, fordi deres vante, trygge skabeloner bryder sammen. Det stiller samtidig store krav til underviseren: At navigere i dette psykologiske spændingsfelt af følelsesmæssig modstand og usikkerhed kræver en dyb psykologisk forståelse og empati for at kunne holde de studerende fast i processen, uden at de mister modet. I tråd med Selvbestemmelsesteorien (Deci & Ryan, 1985) udmønter dette sig ikke i udefrakommende forbud, som blot ville drive brugen af AI i det skjulte; i stedet reagerer de studerende ud fra ægte autonomi og selvregulering. Når de mærker, at automatiske svarmuligheder og ukritiske AI-værktøjer forstyrrer deres egen tænkning og modning, vælger de på eget initiativ at fravælge dem for at opbygge ægte mestring. De teoretiske implikationer viser dermed, at menneskelige udviklingsmodeller med stor fordel kan forklare læreprocesser i programmering, mens de praktiske implikationer – understøttet af de udviklede læringsværktøjer som CodeToModel (Hansen, 2026c), CodeExercises (Hansen, 2026b) og Quiz System (Hansen, 2026d) – giver undervisere et skarpt, realtids-diagnostisk set-up til at spotte og løse de kognitive knudepunkter direkte i klassen. Kort sagt flytter interventionerne undervisningens fokus fra at hjælpe de studerende med at huske den enkelte rute på slottet, til at tvinge dem til at tegne deres eget kort over bygningen.

7. Konklusion

Projektet konkluderer, at der er en direkte sammenhæng mellem studerendes kognitive vanskeligheder med at lære programmering og en manglende eller utilstrækkelig anvendelse af Reality Testing. Ved at implementere den fire-niveaudelte zonemodel og målrettede didaktiske interventioner flyttes de studerende fra usikker gætteadfærd til aktiv virkelighedsprøvning og robust, uafhængig mestring. Projektet understreger dog, at denne

kognitive transformation er krævende og forbundet med ægte modstand for novicen samt et højt krav til underviserens psykologiske dømmekraft i klasserummet, hvilket er afgørende for at modvirke frafald.

8. Erklæring om brug af AI

I forbindelse med udarbejdelsen af dette Working Paper er der anvendt kunstig intelligens (Gemini 3.1 Pro). AI er specifikt blevet brugt som en strukturel og sproglig sparringspartner til at omskrive forfatterens egne noter, observationer og empiri til sammenhængende akademisk prosa. Som forfatter bærer jeg det fulde ansvar for alt indhold i dokumentet, og al AI-genereret tekst og bearbejdning er blevet kritisk gennemlæst, vurderet og verificeret for faglig korrekthed i overensstemmelse med de gældende retningslinjer.

Referencer

- Caspersen, M. E. (2007). *Educating Novices in the Skills of Programming* [Ph.d.-afhandling]. Aarhus Universitet.
- Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. Plenum.
- Enebjörk, M. v. d. D. (2022, 3. februar). *Assignment, scope, and sequence: Code tracing as a tool to improve novice programmers' mental models* [Bachelor's thesis, Uppsala University]. DiVA.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences (PNAS)*, 111(23), 8410–8415.
- Freud, S. (1923). *The Ego and the Id*. Hogarth Press.
- Hansen, J. W. (2026a). *Lektoranmodning* [Unpublished manuscript]. Erhvervsakademi Dania.
- Hansen, J. W. (2026b). CodeExercises [Webapplikation]. Hentet fra https://codeexercise.jakup.dk
- Hansen, J. W. (2026c). CodeToModel [Webapplikation]. Hentet fra https://codetomodel.jakup.dk
- Hansen, J. W. (2026d). Quiz System [Webapplikation]. Hentet fra https://jakup.dk/quiz
- Hastrup, K., Jørgensen, P. S., & Hegland, T. J. (2026). Aktionsforskning. I *Lex* på lex.dk. https://lex.dk/aktionsforskning
- Klein, M. (1932). *The Psycho-Analysis of Children*. Hogarth Press.
- Laplanche, J., & Pontalis, J.-B. (1973). *The Language of Psycho-Analysis*. Hogarth Press.
- Lister, R., Adams, E. S., Fitzgerald, S., Fones, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B., & Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *SIGCSE Bulletin*, 36(4), 119–150.

- Ma, L. (2007). *Investigating and improving novice programmers' mental models of programming concepts* [Doctoral dissertation]. University of Strathclyde.
- Nielsen, K. H. (2026). Hypotetisk-deduktiv metode. I *Lex* på lex.dk. https://lex.dk/hypotetisk-deduktiv_metode
- Norman, D. A. (1983). Some observations on mental models. I D. Gentner & A. L. Stevens (Red.), *Mental models* (s. 7–14). Lawrence Erlbaum Associates.
- Olsen, O. A. (2026). Objektrelationsteori. I *Lex* på lex.dk. https://lex.dk/objektrelationsteori
- Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., & Thota, N. (2019). Pass Rates in Introductory Programming and in Other STEM Disciplines. I *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '19)* (s. 53–71). Association for Computing Machinery. https://doi.org/10.1145/3344429.3372502
- Sorva, J. (2013). Notional machines and introductory programming education. *ACM Transactions on Computing Education (TOCE)*, 13(2), Article 8. https://doi.org/10.1145/2483710.2483713
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285.
- Vaknin, S. (2024). *Gain Self-knowledge via Insight, Let Intuition Protect You* (00:10:39–00:18:19) [Video]. YouTube. https://www.youtube.com/watch?v=CK2xlHGep3g&t=639s. Tilgået 22. juni 2025.
- Xie, B., Nelson, G. L., & Ko, A. J. (2018). An explicit strategy to scaffold novice program tracing. I *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)* (s. 446–451). Association for Computing Machinery.

Bilag 1: Oversigt over de syv undersøgelser

For at tydeliggøre, hvordan hypotesen blev testet i praksis, beskrives formål, metode og centrale fund for de syv undersøgelser herunder:

- **Undersøgelse 1 (Person A, 04.08.2025):**
  - *Formål:* Afdække indledende kognitive barrierer hos en svag studerende gennem en kvalitativ "tænk-højt"-test.
  - *Metode:* Observation under en live-kodnings- og fejlfindingsseance.
  - *Centrale fund:* Totalt fravær af Reality Testing (RT). Den studerendes hukommelse og indre forventning blokerede for computerens objektive skærmfeedback, idet vedkommende påstod, at koden returnerede det forventede resultat trods modsatrettede beviser på skærmen.
  - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 2 (Person A, 11.08.2025):**
  - *Formål:* Indsamle objektive data på den fundamentale forståelse og teste optimeringen af visualiseringsværktøjet CodeToModel (CtM) (Hansen, 2026c).

    - *Metode:* Kvantitative online-quizzer om variabel-tildelinger og "tracing", efterfulgt af praktiske debugger-øvelser.
    - *Centrale fund:* Massive barrierer på det fundamentale niveau, især ved sekventiel sporing af flere variabler over tid, hvilket førte til et redesign af CtM fra "kasser" til visuelle "pegepinde".
    - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 3 (Person A, 18.08.2025):**
    - *Formål:* Tvinge det kognitive arbejdstempo ned og introducere test-cases som virkelighedstjek.
    - *Metode:* Kvalitativ intervention i CodeExercises (CE) (Hansen, 2026b), hvor den studerende skulle definere "input/output"-eksempler før kodning.
    - *Centrale fund:* Test-tænkning var fraværende, og den studerende ignorerede fortsat kodens logik under gennemgang, men interventionen skabte bevidsthed om behovet for løbende verifikation.
    - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 4 (Person B, 19.08.2025):**
    - *Formål:* Undersøge om andre svage studerende udviser samme barrierer.
    - *Metode:* Kvalitativ observation og samtale under opgaveløsning.
    - *Centrale fund:* Udbredt "magisk tænkning". Den studerende argumenterede for, at variabler kunne indeholde udefinerede data ud fra et indre ønske om mening frem for en analyse af den objektive virkelighed.
    - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 5 (Person A og B, 25.08.2025):**
    - *Formål:* Teste effekten af skriftlige forudsigelser (predictions) for at modvirke gætteri.
    - *Metode:* Sammenlignende observation, hvor de studerende blev tvunget til skriftligt at forudsige kodens output, før de måtte køre den i CE (Hansen, 2026b).
    - *Centrale fund:* Fungerede som et afgørende "hastigheds-vejbump", der afskar de studerende fra at rette i blinde og tvang dem til at konfrontere deres fejlbehæftede mentale modeller.
    - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 6 (12 studerende, 11.02.2026):**
    - *Formål:* Teste hypotesen om "hoved-debugging" i stor skala uden computer.
    - *Metode:* Kontrolleret eksperiment, hvor studerende individuelt på tavlen skulle forklare en logisk tricky metode.
    - *Centrale fund:* Stærke studerende brugte spontant konkrete tal til at gennemtvinge test af logikken, mens svage studerende strandede på syntaks. Førte til fjernelse af multiple-choice i QS (Hansen, 2026d).
    - *Fuld rapport: [Se PDF-bilag]*
- **Undersøgelse 7 (Person C, 27.02.2026):**
    - *Formål:* Evaluere den langsigtede effekt af de daglige RT-interventioner.

- *Metode:* Dybdegående kvalitativ case-analyse i en kompleks arkitektonisk opgave, suppleret med 100 % objektiv test-validering i CE (Hansen, 2026b).
- *Centrale fund:* Den studerende opnåede stor kognitiv modning, brød selvstændigt en stor opgave ned i professionelle metoder, fravalgte frivilligt AI/code-completion, og bestod den uafhængige test fejlfrit.
- *Fuld rapport: [Se PDF-bilag]*