Understanding ADHD Developers' Challenges Through the Lens of Developer Experience: A Qualitative Study of the r/ADHD_Programmers Community on Reddit

Natalya Goelzer
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
natalya.goelzer@edu.pucrs.br

Sofia Sartori
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
sofia.batista@edu.pucrs.br

Tatiana Cartagena
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
cartagena.tatiana@edu.pucrs.br

Karina Kohl
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
karina.kohl@inf.ufrgs.br

Gabriel Velloso
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
velloso.g@edu.pucrs.br

Sabrina Marczak
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
sabrina.marczak@pucrs.br

Abstract

Developer Experience (DevEx) frames software development around three dimensions: flow state, cognitive load, and feedback loops. Existing research often treats developers as cognitively homogeneous, leaving unclear how these dimensions are experienced by developers with Attention Deficit Hyperactivity Disorder (ADHD). We address this gap through a reflexive thematic analysis of 241 comments from r/ADHD_Programmers, a peer-support community offering rich accounts of daily software work. Our findings recast each dimension: flow is less a state developers enter than a fragile resource they must protect, through an ambivalent hyperfocus sustained only with team support; cognitive load is driven less by task complexity than by the gap between demand and available executive capacity, where unclear requirements trigger executive paralysis rather than merely slowing work; and feedback loops act as external supports for attention and memory, where reliability and persistence matter as much as speed. Together, these suggest the DevEx dimensions remain useful but are linked by a shared executive-function base, pointing to ADHD developers' strategies as candidate process improvements for the broader developer population, not only individual accommodations.

Keywords

Developer Experience, Neurodiversity, Cognitive Load, Flow, Feedback Loops

1 Introduction

Developer experience (DevEx) treats software development as an intellectual activity that depends on both thought and motivation, offering a lens for analyzing and improving project environments as perceived by developers [19]. Building on this subjective framing, Noda et al. [34] operationalized the concept through three dimensions: feedback loops, cognitive load, and flow state. These dimensions aim to optimize the timeliness of development signals, reduce mental effort, and foster conditions for immersive, productive focus.

However, most DevEx literature treats the developer as a largely homogeneous cognitive subject, assuming stable attention, predictable working memory, and effort regulation that responds linearly to environmental adjustments. This implicit neurotypical baseline becomes consequential given that up to 20% of the population is estimated to be neurodivergent [16], and that the 2024 Stack Overflow Developer Survey reports 10.27% of respondents with a concentration and/or memory disorder such as Attention-Deficit/Hyperactivity Disorder (ADHD) [45]. ADHD, in particular, is associated with persistent differences in attention regulation, working memory, and executive functioning [1, 5, 15], precisely the capacities that the three DevEx dimensions presuppose.

Software engineering research has begun to explore this space. Studies have documented challenges related to attention, time management, communication, and task alignment, as well as strengths such as creativity and hyperfocus under supportive conditions [33]. Complementary research has shown how these experiences emerge from the interaction between executive-function differences, social dynamics, and organizational structures, highlighting the role of both personal strategies and workplace adaptations [22, 27]. Collectively, these findings suggest that the experience of developers with ADHD is shaped by the interplay between individual regulation and socio-technical context. However, these studies organize findings around challenges, strengths, or adaptations, often at the level of tasks, practices, or coping strategies, rather than the broader experience of software work itself.

What remains less understood, then, is whether the three dimensions of DevEx [34] remain operative for ADHD developers, and how their content is amplified or reshaped when the underlying cognitive infrastructure is variable. In particular, these dimensions have largely been framed assuming relatively stable patterns of attention and effort regulation. When these assumptions do not consistently hold, it becomes less clear how such dimensions are experienced in practice, and whether they capture the dynamics of software work in the same way.

To address this gap, this study investigates how ADHD developers perceive and navigate the three core dimensions of DevEx in everyday software work. We conducted a thematic analysis [8] of 241 comments from r/ADHD_Programmers, a Reddit community where programmers with ADHD exchange experiences. The comments were coded to identify recurring patterns and then interpreted in relation to the three DevEx dimensions. Our findings suggest that the structure of DevEx remains operative for ADHD developers, but the content of each dimension is amplified and reshaped by executive-function demands: flow appears as a state to be defended against competing attentional demands rather than only entered; cognitive load is regulated less by reducing task complexity than by externalizing executive effort onto artifacts and rhythms; and feedback loops act not only as latency-reducing technical mechanisms but as external regulators of attention. By organizing empirical accounts within DevEx dimensions and connecting them to executive-function patterns, this study offers an initial articulation that allows evidence on ADHD developers to be interpreted within a framework that already informs industry decisions about how software work is designed.

2 Background and Related Work

2.1 Developer Experience

Developer Experience (DevEx) treats software development as an intellectual and affective activity shaped by cognition, motivation, emotion, and intentional action [19]. It frames development experience not as productivity or tool performance alone, but as the interaction between developers, tasks, tools, processes, and work context, operationalized through three interacting dimensions: feedback loops, cognitive load, and flow state [34]:

- **Flow state** refers to a condition of sustained and purposeful engagement in which developers can remain focused on a task without avoidable disruption.
- **Feedback loops** refer to how quickly and clearly developers receive signals about whether their work is progressing correctly, including signals from builds, tests, code reviews, stakeholders, and teammates.
- **Cognitive load** refers to the amount of information developers must hold, interpret, and coordinate while working, which can be affected by unclear requirements, context switching, tool friction, and fragmented communication.

These dimensions are often interdependent: delayed feedback or high cognitive load can impede flow, while streamlined feedback and clear goals may facilitate deep work [20]. Yet most DevEx research has examined developers without differentiating cognitive profiles [29, 36, 46], a limitation for work experiences shaped by variation in attention, memory, and regulation [45].

2.2 Attention-Deficit/Hyperactivity Disorder

Attention-Deficit Hyperactivity Disorder (ADHD) is defined by persistent patterns of inattention and impulsivity that affect executive control and motivation [1]. Beyond diagnostic symptoms, it is described as a cognitive and regulatory profile marked by differences in executive control, motivation, and reward processing [41]. These differences are often framed through Executive Functions (EF): Brown [10] organizes executive functioning into six clusters that connect ADHD-related mechanisms to observable work patterns:

- **(1) Activation**: planning and starting tasks (organizing, prioritizing, estimating duration, initiating), often linked to procrastination or trouble beginning tasks that do not seem immediately interesting or urgent;
- **(2) Focus**: maintaining and redirecting attention amid distractions, ranging from easy distractibility to hyperfocus, where attention becomes too intense to switch away from;
- **(3) Effort**: sustaining mental energy, alertness, and a consistent work pace, with difficulties in repetitive or low-stimulation activities and fluctuating productivity over time;
- **(4) Emotion**: managing frustration and sensitivity to feedback or criticism, where strong emotional responses may temporarily interfere with concentration and decision-making;
- **(5) Memory**: holding and using information while working, with difficulties such as forgetting steps, losing track during discussions, or connecting new information to prior knowledge;
- **(6) Action**: monitoring and adjusting behavior in real time, including inhibiting impulsive reactions and adapting actions to the situation or environment.

ADHD is also context-sensitive: engagement can be high when tasks are urgent or novel, yet friction rises in environments with frequent interruptions or low structure [26, 43]. This sensitivity is relevant to software engineering, which requires sustained focus, working memory, and problem decomposition [3, 44]. Consequently, the DevEx dimensions of flow, load, and feedback provide a framework for examining how software work environments interact with an ADHD cognitive profile.

3 Research Method

This section presents our research strategy, covering the research questions, data collection and selection procedures, and the qualitative analysis, with an emphasis on methodological rigor and transparency.

3.1 Research Questions

To investigate how the core dimensions of Developer Experience [34] are perceived by developers with ADHD, we propose the following research questions (RQs):

- **RQ1. Under what personal, task, and team conditions do ADHD developers enter, sustain, or lose flow during development?** This question aims to characterize the dynamics of the flow state in this population.
- **RQ2. What sources of cognitive load arise across tools, processes, and coordination, and which conditions mitigate or intensify this load?** This question maps the sources of mental effort that ADHD developers describe as particularly tough in daily work.
- **RQ3. What properties of feedback loops do ADHD developers report in everyday work, and with what effects on attention regulation and task progress?** This question investigates how the quality of these loops influences attention regulation and the continuity of task execution.

3.2 Data Collection

To address our Research Questions we employed thematic analysis (TA) following Braun and Clarke's[8] six-phase framework, an approach well-suited for interpreting rich, naturalistic qualitative data from online communities. We analyzed public discussions from *r/ADHD_Programmers*, the largest open forum dedicated to ADHD programmers ($\approx 82,000$ members, $\approx 27,000$ weekly visitors as of August 2025), to understand the lived experiences of ADHD developers.

We chose it because long-form, threaded discussions and pseudonymous participation create a safe space for candid disclosure about neurodiversity [32, 40]. Unlike Stack Exchange and similar Q&A platforms [12, 42], the subreddit is oriented toward peer support, where practitioners share opinions, feelings, and everyday challenges more openly, yielding rich, real-world evidence.

We collected data with the Python Reddit API Wrapper (PRAW) [6] under authenticated read-only access, complying with Reddit's Data API and Public Content policies [37, 38]. From the "Top/All" listing for the first half of 2025, we retrieved each post's full thread (all comments and replies) using *replace_more(limit=None)* so that no responses were omitted, filtering locally by UTC creation timestamps.

3.3 Data Selection

To refine the initial dataset into the final analytic sample, we applied a two-stage selection process guided by predefined inclusion criteria (IC), detailed in Table 1.

Table 1: Inclusion Criteria and their underlying logic in the selection process.

ID	Inclusion Criterion	Logic
IC1	The thread must contain at least one comment.	Structural validity
IC2	The thread must address experiences related to software development work.	Topical relevance
IC3	The thread's Score must be greater than 400.	Popularity and community validation
IC4	The content must match keywords from the DevEx/ADHD lexicon.	Lexical alignment with research scope
IC5	The content must include experiential elements (emotions, strategies, impacts).	Qualitative depth and analytical density

Stage 1: Quantitative and Lexical Filtering. The initial dataset, obtained via Reddit scraping, comprised 115 threads and 1,441 comments. We retained threads with at least one comment (IC1) addressing software development experiences (IC2), and applied a popularity cutoff of Score > 400 (IC3) as a marker of community relevance. A lexical filter (IC4) then kept content matching a keyword lexicon of DevEx and ADHD/executive-function indicators (e.g., *hyperfocus*, *working memory*, *time blindness*, *cognitive load*, *context switching*, *flow*, *procrastination*), drawn from preliminary corpus inspection and prior work [32, 33]. This reduced the dataset to 25 threads and 820 comments.

Stage 2: Qualitative Review and Analytical Density. We then reviewed the 25 candidate threads by Score, comment volume, and textual density. To reduce bias, two researchers independently read each thread end-to-end, judging whether it offered personal narratives tied to at least one DevEx dimension; divergences were resolved by consensus. Threads were excluded when off-topic, self-promotional with empty bodies, or saturated by content already covered elsewhere. Within retained threads, comments lacking analytical density (e.g., emojis, brief agreements, short confirmations) were also excluded (IC5). This yielded a final sample of 241 comments from 10 threads. To protect privacy, all usernames were replaced with sequential pseudonyms (e.g., *U_1*), with a stable mapping across posts from the same user.

3.4 Qualitative Data Analysis

We applied reflexive TA following Braun and Clarke's six-phase framework [8, 9, 11], from a contextualist, experiential position. The DevEx dimensions served as sensitizing concepts orienting the initial reading, but coding stayed predominantly inductive: codes were generated from the data rather than a pre-existing frame, and themes were actively constructed through sustained engagement with the corpus rather than treated as pre-existing features to be found [9]. Coding operated at both semantic and latent levels, capturing explicit content and underlying assumptions.

Before starting, two researchers calibrated by independently analyzing the same three threads and discussing extracted segments and emerging codes to align interpretations and reduce bias. The analysis then proceeded through the six phases:

(1) **Familiarization with the data:** All authors read the full sample (241 comments) at least once before coding and forming judgments, then iteratively re-read each thread end-to-end, attending to the comment-tree structure and recording reflexive memos.
(2) **Generating initial codes:** We systematically labeled relevant features across the corpus, using descriptive codes for explicit content and interpretive codes (e.g., Burnout sensitivity reveals systemic issues; Double-edged appraisal of ADHD traits) for underlying meanings. The unit of analysis was typically a sentence or clause expressing one idea, and multiple codes could apply to the same extract. Following reflexive TA guidance, accounts that departed from the dominant story were retained rather than discarded.
(3) **Searching for themes:** Codes were iteratively clustered into higher-level codes and assembled into candidate sub-themes, provisionally placed within the three DevEx dimensions. Two researchers built an initial thematic map across several sessions, rearranging codes to test alternative groupings.
(4) **Reviewing themes:** Candidate themes were reviewed against dual criteria: internal coherence against collated extracts (data within a theme cohering meaningfully) and external distinctiveness across themes. Each grouping was tested against

a central organizing concept that articulated its shared meaning, which served as the primary artifact for resolving disagreements and keeping the thematic structure auditable.

(5) **Defining and naming themes:** For each theme and subtheme, we articulated a central organizing concept and finalized concise, communicative names, updating the thematic map and code-theme log.

(6) **Producing the report:** The results section reports findings as an analytic narrative organized by research question, with extracts chosen to illustrate each claim and prioritizing vividness and representativeness.

Final Codebook. The analysis produced 362 codes, consolidated into 33 sub-themes and 15 themes across the 3 DevEx dimensions; the full codebook and thematic map are in the replication package [23]. Consistent with reflexive TA, we did not compute intercoder agreement, since this approach treats coding as interpretive rather than the recovery of fixed meanings [9]. Methodological quality was instead pursued through prolonged engagement, calibration meetings, reflexive dialogue among authors, and an audit trail linking code to thread and comment [13].

4 Results

This section presents the findings from our reflexive thematic analysis of comments collected from r/ADHD_Programmers.

4.1 RQ1: Dynamics of Flow State in ADHD Developers

Flow refers to deep and productive immersion supported by clear goals, appropriate challenge, and limited interruptions [34]. Figure 1 summarizes the six themes that, in our reading, help explain how ADHD developers entered, sustained, or lost flow.

Attention dynamics. The data suggests flow was not described as a stable trait, but as unstable transitions between attention, distraction, and deep immersion. In Focus & attention shifts, participants described flow loss not as a lack of interest in programming, but as difficulty sustaining task direction until completion: *"I do like coding. I struggle a lot with finishing tasks or focusing without procrastinating"* (U1). Attention was portrayed as non-linear(*"brain just won't think in a straight line"* (U33) ,*"Usually I have a moment of self consciousness and that's enough to lose it"* (U78)). Visible task structure appeared to help developers re-anchor attention, while its absence made channeling effort harder: *"I'm full of energy but can't channel it productively"* (U82).

Entering hyperfocus or flow appeared as an interest-dependent state supported by routines, energy alignment, and task fit, rather than as a fully voluntary condition. Participants described preparation strategies, such as *"Getting hyperfocus on-demand… cappuccino, music, stretch… mental 'trigger' varies with rest/cycle"* (U20), and reported long immersion when tasks matched their interests: *"I can sit down for 5–6 solid hours a day of paid hyper fixation dives"* (U87). Flexible work arrangements could enable flow by aligning work with personal energy windows (*"Work from home… make up for lack of focus at night"* (U20)), though hyperfocus also appeared fragile and not fully controllable. To extend engagement, participants described Mode-switching to sustain engagement, as in *"some code + strategy and solution search works for me"* (U26), while Sustaining and exiting hyperfocus suggests the friction was not only entering immersion, but recovering when energy collapsed: *"On those days I just can't… I help on Slack to get out of an ADHD crash"* (U87).

Interruption emerged as one of the main conditions disrupting flow because, in the data, it did not only pause activity but broke the continuity needed to resume it. In Unplanned interruptions & reactive work, interruptions were described as events that destroyed task state and increased recovery effort: *"break my flow"* (U20); when stopped mid-task, *"I wouldn't have been able to pick up where I was at"* (U79). The accounts suggest this intensified in reactive operational work; in on-call situations, one participant described being *"interrupted from one urgent ticket by being paged for another"* (U117), leaving *"the only choice"* (U117) of deciding which task to neglect. Mitigations therefore included not only individual focus aids but also mechanisms preserving task state, such as creating *"a task for every ticket, together with subtasks and notes"* (U48).

In Context switching & multitasking, switching appeared harmful when imposed but potentially productive when controlled. The friction emerged when switching exceeded working-memory capacity, as in *"reading my own code is a nightmare"* (U150) which led developers to *"start from 0 each time"* (U150); initiation itself was described as difficult: *"starting tasks is brutal with ADHD"* (U23). Yet some participants transformed switching into a strategy by keeping *"multiple projects ready"* (U157) so that attention shifts moved toward something valuable rather than unrelated distractions.

In Notifications & ambient distractions, alerts were described as unstable forms of support: useful at first, but often becoming too disruptive or too easy to dismiss (*"swipe away/ignore everything"* (U186); *"swiping away the notifications become second nature"* (U188)). The data suggests more effective strategies involved persistent, socially-supported cues like a *"taskboard that stays visible, not just another notification"* (U189) and *"body doubling and multiple weekly checkins"* (U188). In Meetings & scheduled interruptions, even planned events fragmented the workday by creating anticipatory waiting: a late standup could place the developer in *"waiting mode"* (U129) for the whole day. Days with *"one big task from 9 to 5"* (U132) or *"peaceful 9–5 hyperfocus"* (U139) were described as preferable, suggesting flow depended less on isolated accommodations and more on temporal stability and visible task structures.

Motivation & meaning appeared shaped by the fit between task, developer interests, and perceived value. In Meaningful work & engagement, coding alone was not described as motivating; engagement depended on challenge, novelty, autonomy, and visible progress, since ADHD brains *"crave stimulation"* (U25) and *"dopamine rewards for progress"* (U184) could sustain persistence. When this alignment was absent, even enjoyable technical work became draining: *"you like coding but you're not actually doing anything meaningful, interesting, or challenging enough and so you start to burn out"* (U2). Paid-work constraints reduced motivation when developers were *"stuck doing it on someone else's timeline, with their decisions, and at their direction"* (U17). Conversely, motivation seemed to increase when developers could shape work around curiosity and strengths: being *"more of a generalist"* (U2), moving on *"by the time the novelty wears off"* (U4), or going *"down a rabbit hole of new info"* (U87).

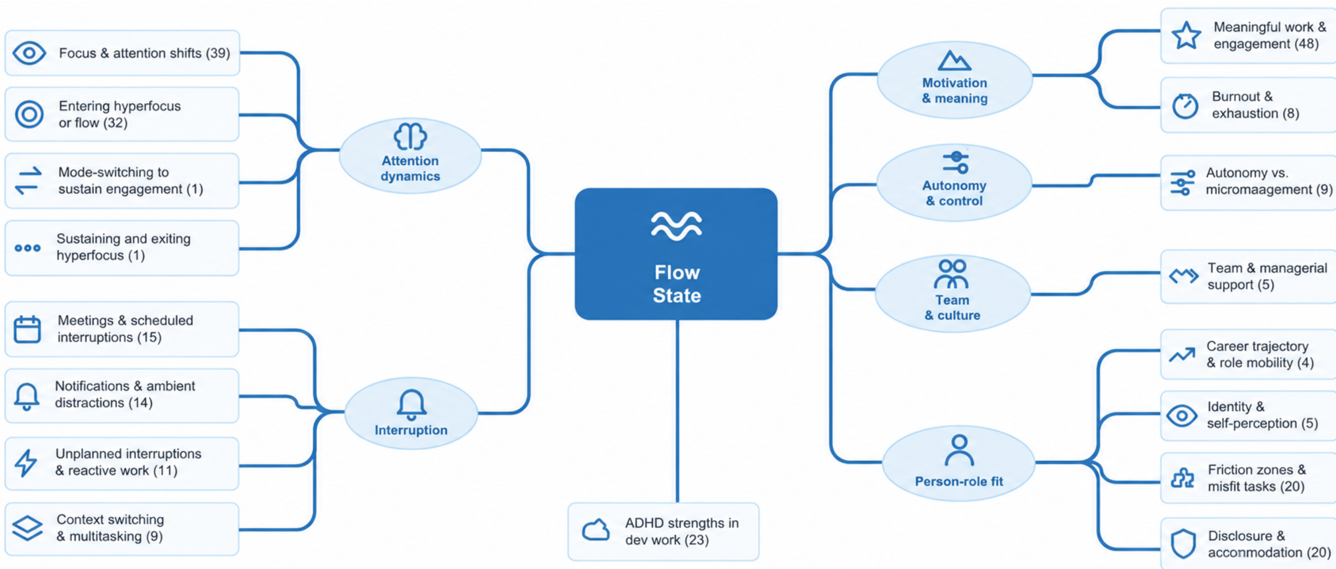

Figure 1: Conceptual map for RQ1: conditions that enable, sustain, or disrupt flow.

When meaningfulness was absent for long periods, the accounts suggest depletion rather than simple boredom. In Burnout & exhaustion, fatigue from repeatedly losing and rebuilding task context was summarized as *"I am so cooked"* (U65); at the work-system level, exhaustion intensified when concerns had *"fallen on deaf ears"* (U160) and experienced *"permanent state of exhaustion"* (U160). Some participants connected ADHD burnout to broader productivity norms, framing ADHD developers as a *"canary in the coal mine"* (U6) for unsustainable work conditions.

Person-role fit shaped, in the data, when ADHD-related traits supported or disrupted flow. In ADHD strengths in dev work, programming was not framed as inherently suitable or unsuitable; the central issue was whether *"the type of programming or job isn't the right fit"* (U5). Flow appeared easier when roles offered stimulation, autonomy, and difficult problem-solving: being *"more involved in scoping, designing, architecting solutions"* (U4) or working on *"the enormous projects no-one else dares to touch"* (U149), where the *"constant need of intellectual stimulation"* (U149) appeared productive rather than disruptive.

Across Career trajectory & role mobility, Identity & self-perception, and Disclosure & accommodation, sustaining flow seemed to depend on learning, negotiating, and protecting this fit over time. Better-fit roles often appeared later in the career, since senior autonomy *"requires that you get through the junior roles yourself first"* (U4). Self-understanding helped developers *"recognize and accept your approach…and not fight it"* (U15), and difficulties were framed as not the discipline but as a function of whether environments were *"more accommodating than others"* (U13). Flow appeared more sustainable when disclosure led to support, as in the managerial response *"What can I do to help?"* (U106). By contrast, in Friction zones & misfit tasks, flow was lost when role demands conflicted with attention or motivation regulation, as illustrated by the contrast between *"write code all day based on PRD I'd go nuts"* (U26) and work combining *"some strategy and solution search"* (U26).

Autonomy & control shaped whether ADHD developers could adapt work to their attention patterns. In Autonomy vs. micromanagement, flow was enabled by trusted latitude to engage with meaningful, complex work, that is, *"collaborative"* (U5) problems that seem *"too big"* (U5) to reach *"peak effectiveness"* (U5), and disrupted when autonomy was replaced by control: *"I wither when I'm micromanaged or expected to do repetitive, solved work"* (U5).

Team & culture shaped, in our reading, whether flow was sustained as an individual effort or collectively supported by the surrounding social system. In Team & managerial support, peer and managerial guidance appeared to reduce ambiguity and ease re-entry after interruptions: *"An accountability buddy helps me follow through"* (U29); peers may have *"a better idea of how to make progress in a more efficient way"* (U33). By contrast, flow was disrupted when developers managed uncertainty without support, especially when *"forced to triage without visibility or authority"* (U120), framed as a management failure rather than a personal one.

4.2 RQ2: Sources and Mitigation of Cognitive Load

Cognitive load represents the mental effort required to perform complex tasks, where sustained high levels can compromise productivity and increase stress [34]. Building on this premise, Figure 2 synthesizes the analyzed data and the five themes that illustrate how such load was produced, intensified, and mitigated within our study.

Requirement clarity appeared to determine whether attention could go directly to development or instead be spent reconstructing

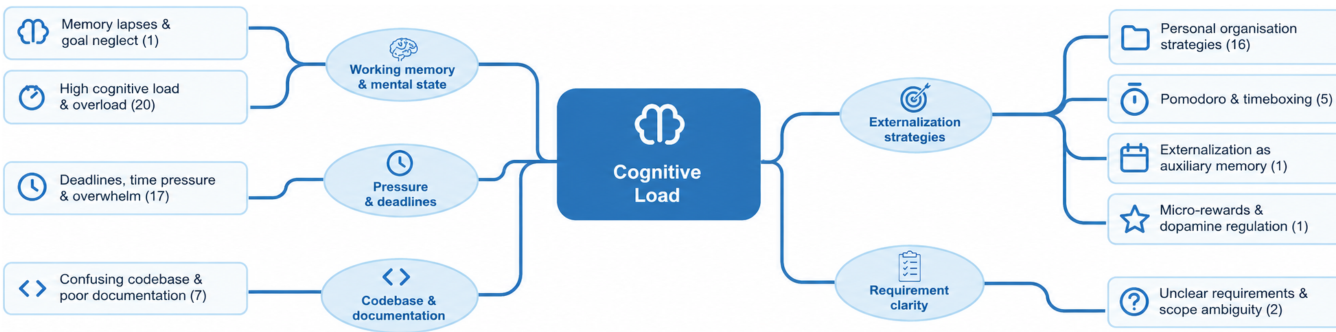

Figure 2: Conceptual map for RQ2: sources and mitigators of cognitive load.

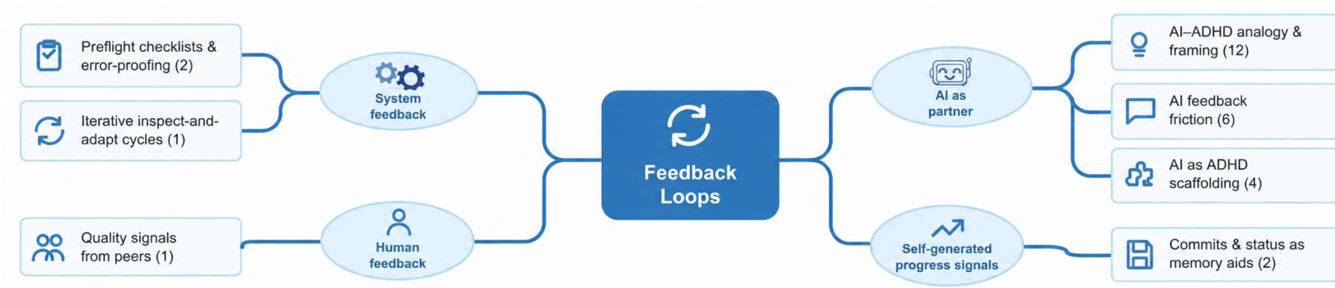

Figure 3: Conceptual map for RQ3: feedback properties and effects on attention regulation and progress.

goals and completion criteria. In Unclear requirements & scope ambiguity, load seemed to be mitigated when work offered *"a constant stream of things to do, a well defined definition of done... and you're given the time you need"* (U6). When scope was less defined, broad experience could partially compensate, as a *"very broad tech radar and approximate knowledge of many things"* (U8) helped developers navigate uncertainty during early-stage work.

Codebase & documentation extended this load to technical navigation. In Confusing codebase & poor documentation, the burden involved reconstructing paths, assumptions, and prior decisions, described as a *"backtracking algorithm, like solving a maze: you choose a path, and if you hit a dead end, you return... to the last decision point and try a different route"* (U35). The contrast between being able to *"create an IMS online program in one day"* (U108) and the slower work of *"maintaining someone's 10-20yo code"* (U108) suggests that load increased when the technical context was old, opaque, or poorly documented.

Working memory & mental state appeared to set the baseline cost for cognitive load, since developers had to hold task goals, partial decisions, and interruptions in mind while coding. In High cognitive load & overload, working memory was described as limited and easily displaced: *"I only have few carriages... the active convo taking all of my RAM space has to unload sometimes critical bits"* (U62). Participants noted minds *"tend to both get fixed on one thing and sometimes forget things we've already thought about"* (U34), leading to repeated solution paths. This profile was not framed as only deficit-based: *"your extreme pattern recognition kicks in and your ability to laterally think starts paying dividends"* (U36).

In Memory lapses & goal neglect, overload became more disruptive when developers kept working but lost the original target, as in trying to *"bruteforce all possible paths to the mountain"* (U37), only to realize they were *"meant to go the valley because I forgot what I pointed at to start off with"* (U37). Mitigation depended on making work state visible outside memory: doing design before coding to avoid *"going too deep into actually writing code, whilst actually figuring it all out"* (U34), using a *"memory map"* (U183) for a *"bird's-eye view of software architecture"* (U183), or keeping a *"physical Kanban-Board hanging above my desk"* (U187).

Pressure & deadlines shaped cognitive load by turning time into both an activation cue and a source of overload. In Deadlines, time pressure & overwhelm, participants described a cycle in which low productivity increased anxiety until urgency enabled action: *"My ADHD messes with my productivity. The productivity (lack thereof) messes with my anxiety. The actual deadline makes me get shit done"* (U89). Pressure could narrow attention, as in *"a tough deadline asking me to do a presentation"* (U111), but also forced developers to abandon preferred strategies (*"on time crunches, I cant brute force forever sadly"* (U38)). Over time, this seemed to make work depend on high-stakes activation, with some becoming *"the firefighter, the go-to person for high stakes... kind of stuff in critical situations"* (U98).

Externalization strategies formed, in our reading, the main mitigation pattern by moving task state, decisions, and motivation cues out of working memory. In Personal organisation strategies, participants reduced initiation cost by *"breaking it into micro-tasks that can be done in 5–10 mins"* (U23); agile structures appeared as the team-level analogue, where *"User stories and tickets are broken down into sub-tasks… no more than 1–2 days"* (U143), creating *"regular "done" moments every couple of days"* (U143). In Externalization as auxiliary memory, written artifacts helped preserve reasoning, as in the need to *"write down and do design… as it both stops you re-thinking things and it's less effort costly than actually writing code"* (U34). Time-based strategies operated similarly: in Pomodoro & timeboxing, participants externalized mental noise by listing *"everything ruminating in my head"* (U90) and worked in short windows because *"it's just 25 mins. Everything else can wait for 25 mins"* (U90).

In Micro-rewards & dopamine regulation, externalization also involved regulating motivation through small visible rewards, as in building *"periodic wastes of time" into my schedule"* (U157) tied to *"micro-achievements"* (U157). Participants also emphasized that no isolated tool was sufficient: *"no single tool/app" – no matter how good it is – will be enough on its own in practice"* (U188).

4.3 RQ3: Feedback Loops and Attention Regulation

Feedback loops refer to the speed, clarity, and continuity of responses from tools, processes, and collaborators [19, 34]. In the analyzed data, feedback loops appeared as mechanisms for information exchange and as external regulators of attention that helped ADHD developers maintain context, reduce uncertainty, and identify the next action. As summarized in Figure 3, four themes describe how feedback loop properties supported or disrupted attention regulation.

AI as feedback partner emerged as the densest theme in this dimension, suggesting ADHD developers interpreted AI tools not only as coding assistants but as feedback partners with both enabling and disruptive effects. In AI–ADHD analogy & framing, developers framed these tools through an ADHD analogy, describing AI coders as fast but poorly regulated: *"AI coders, like cursor, feels very much like ADHD. It's like having a ferrari engine with toyota brakes"* (U40).

In AI as ADHD scaffolding, AI feedback appeared useful when it converted abstract or unfinished work into visible next steps, supporting planning, refactoring, visualization, and continuation after attention drift. One developer used AI to *"make me a timeline doc that it auto-updates as I go and gives me options for what to work on next"* (U68), positioning AI as a continuity aid for choosing the next action. Another described Cursor as taking over cleanup: *"Cursor just refactors it, letting me do the fun work of solving the problem and then taking over when my ADHD won't let me clean it up"* (U42). AI also supported external representation, as in using Draw.io with an LLM when the developer had *"a clear idea I can put in writing"* (U126) but was unsure how to draw it. Together, these accounts suggest helpful AI feedback worked as an executive-function scaffold.

In AI feedback friction, however, participants described feedback loops that were fast but unstable, forgetful, or too costly to supervise. AI could be engaging because it could *"switch direction on a dime"* (U44), but this same property was *"bad for my focus tho"* (U44). Friction also emerged when tools failed to preserve prior instructions: *"tell it the same ground rule multiple times and still it forgets after a while"* (U45); *"forget to fix their mistake after you explain it"* (U46). The accounts suggest this reversed the expected value of feedback: instead of reducing uncertainty, the loop generated repeated correction work. Participants also raised a motivational concern, as AI may remove part of the exploratory pleasure of development: *"one just asks and the answer is served on a platter without much exploration"* (U181).

Self-generated progress signals suggest that ADHD developers created feedback loops for themselves when external signals were absent or too weak to preserve task continuity. In Commits & status as memory aids, commits were repurposed as memory anchors. One participant described writing the commit message before implementation *"so when I forget I can quickly look up what my intention was"* (U43). Another recommended *"constant commits even on non working code"* (U66), because then *"you don't have to remember, just scan through the changes you've made and it will bring you back to what you were doing"* (U66).

System feedback appeared to support attention regulation when tool-mediated signals were simple, visible, and available at the moment of action. In Preflight checklists & error-proofing, checklists were described as a way to turn implicit process knowledge into reusable prompts: *"plan with checklists. Build and maintain a library of checklist templates"* (U127). Simplicity also mattered, since systems should provide *"no hidden lists, no complex setup"* (U184), as configuration-heavy tools may fail precisely when executive function is already overloaded.

Human feedback complemented self-generated and system feedback by anchoring developers within social loops of alignment, correction, and validation. In Iterative inspect-and-adapt cycles, short feedback cycles seemed to reduce drift by catching problems early: *"reviews and check-ins catch issues early. You don't spiral down the wrong rabbit hole unnoticed"* (U143). In Quality signals from peers, feedback also appeared through its absence: as described in *"EXTREMELY detail-oriented"* (U152), which led to *"very little to no QA feedback coming back to me because I already thought of the little details and covered my bases"* (U152).

5 Discussion

We discuss how the three dimensions interact (Section 5.1), how the findings relate to prior work (Section 5.2), and what implications emerge for research and practice (Section 5.3).

5.1 ADHD Developer Experience

ADHD-related challenges are documented in software engineering [21, 22, 27, 31, 33] and in clinical literature [5, 15, 25]. Our contribution is to articulate them within the three DevEx dimensions [20, 24, 34], which provide the operational vocabulary used in industry decisions about tooling, workflows, and team practices [2, 7, 47]. Table 2 summarizes the empirical patterns identified in this study; the paragraphs below read them against prior literature, labelling each comparison as converging with, qualifying, or extending existing DevEx and ADHD framings.

The data suggest that the DevEx structure remains usable for ADHD developers, but each dimension's impact is amplified by executive-function demands: frictions that read as inefficiencies for

Table 2: Dimensions, frictions, and strategies related to developer experience.

Dimension	Frictions & Challenges	Enablers & Personal Strategies	Predominant EF
Flow State	**Attention drift**: loss of context and high recovery cost after distractions. **Interruption ecology**: unplanned calls or notifications that break deep immersion. **Task misalignment**: struggle with monotony or math-heavy tasks. **Burnout**: collapse of motivation due to sustained regulatory effort.	**Intrinsic Meaning**: engagement triggered by novelty and purpose. **Hyperfocus**: intense concentration as productivity engine. **Environment Control**: rituals and deep work blocks. **Team Support**: psychological safety and accountability.	**Activation & Focus** (initiation, sustained attention)
Cognitive Load	**Information Overload**: difficulty holding multiple code states. **Ambiguity**: unclear requirements increasing planning effort. **Working Memory Lapses**: forgetting goals mid-task. **Time Pressure**: deadlines increasing perceived load.	**Externalization**: diagrams and notes to offload cognition. **Micro-tasking**: breaking work into small steps. **Time-structuring**: Pomodoro rhythm. **Micro-rewards**: regulating motivation.	**Memory & Effort** (working memory, processing pace)
Feedback Loops	**AI Friction**: tools that repeat mistakes or lose context. **Wait Latency**: slow pipelines causing attention drift.	**AI Scaffolding**: delegating detail-oriented work. **Intent Anchors**: frequent commits to resume focus. **Checklist Templates**: preventing omissions.	**Memory & Action** (intent recall, real-time self-regulation)

neurotypical developers become continuity barriers when working memory, attention regulation, and task initiation are constrained [5, 10, 15]. Recent empirical work concurs, linking pull-request delays to "bad days" [35] and cognitive load to physiological strain [7]. To make this amplification visible, we read each dimension through Brown's six executive-function clusters [10]: Activation, Focus, Effort, Emotion, Memory, and Action.

Cognitive load read through Memory and Effort. Cognitive load was shaped less by absolute task complexity than by the gap between executive demand and available capacity. Working-memory limits (Memory) and uneven mental pace (Effort) [15] made ambiguous requirements and opaque codebases especially costly, widening this gap. The strategies in Table 2 converged on one operation—redistributing executive effort onto external artifacts and rhythms—suggesting load reduction depends as much on sustained externalisation as on tool or requirement design.

Feedback loops read through Memory and Action. Feedback frictions operate not only technically but cognitively, through the cost of holding intent across waiting periods and self-regulating in real time when signals are unstable [19, 34]. This may explain why fast-but-forgetful AI tools add supervisory effort, and why participants relied on commits as memory anchors and on persistent, socially supported cues. Feedback value, then, may depend on reliability and persistence as much as on speed.

Cross-dimensional dynamic. Read together, the dimensions form a regulatory chain rather than independent effects, with three recurrent configurations. In a *loss* configuration, lengthened feedback loops force context reconstruction and raise re-entry cost from interruptions, increasing load and depleting the resources that protect flow. In a *support* configuration, short clear loops, explicit completion criteria, externalised state, and bounded autonomy lower load and channel hyperfocus to valuable work. In a *mixed* configuration, the same mechanism helps or harms by context: parallel tracks absorb idle time but may dilute depth; hyperfocus accelerates clear paths but can lock attention when sequencing is unclear. Two inferences follow. First, strategies migrate across dimensions because the executive resource is shared—frequent commits act as flow protectors, checklists as feedback substitutes. Second, an intervention in one dimension may affect the other two by releasing executive capacity rather than solving a local problem.

5.2 Comparison with Related Work

Our findings corroborate, qualify, and sometimes extend prior literature on neurodivergent software engineers and on general DevEx. We organize the comparison by DevEx dimension, noting whether our interpretation *converges with*, *aligns with*, or *expands upon* prior literature.

Flow and the dual nature of hyper-focus (qualifies prior framings). Conditions for entering flow (uninterrupted time, clear goals, autonomy, adequate challenge) converge with mainstream DevEx [14, 29, 34] and with neurodivergent work linking hyperfocus to interest alignment [27, 31, 33].

Where mainstream DevEx treats flow as a positive state to be engineered through environmental controls [20, 30, 34], our analysis, consistent with prior ADHD-focused studies [22, 25, 27], finds hyper-focus interest-dependent, partially involuntary, and capable of both productive immersion and post-engagement exhaustion. These findings are consistent with recent psychological research suggesting that hyper-focus and flow are related but distinct constructs, with inconsistent empirical correlations between them [4, 18]. For this population, flow is an ambivalent state whose maintenance depends on team-level support.

Interruption ecology and context switching (qualifies prior framings). Our findings converge with prior work identifying interruptions, fragmented schedules, and reactive operational work as drivers of negative experience [7, 22, 27–29, 35]. Two nuances emerge: some participants describe controlled switching among parallel projects as productive, contrasting with framings that treat switching strictly as disruption [7]; and the data suggest habituation to digital notifications and a preference for persistent, socially-supported cues, whereas parts of the literature emphasize active blocking as the dominant adaptation [22].

Cognitive load, ambiguity, and externalization (extends prior framings). The data align with mainstream DevEx in identifying unclear requirements and fragmented workflows as load sources [7, 34], and the externalization strategies resonate with neurodivergence literature [27, 33]. We extend these accounts in two ways. First, ambiguity here is not primarily an efficiency drain; it triggers *regulatory failures* such as executive paralysis and goal neglect, connecting load to executive function rather than information-processing cost alone. Second, the externalization repertoire is not purely individual: agile structures (short-cycle user stories, frequent "done" moments) act as team-level externalization devices, and micro-rewards function as deliberate motivation regulators linking work practices to clinical compensatory strategies. Notably, some widely cited adaptations such as physical artifact visibility were less salient in our data, suggesting this population may rely more on digital and process-integrated cues—a pattern warranting further investigation.

Feedback loops as external regulators of attention (extends prior framings). While mainstream DevEx focuses on latency reduction [2, 20, 34] and neurodivergence studies often treat feedback as an isolated reward or communication mechanism [27, 33], we position feedback loops as *external regulators of executive function.* Three points support this. First, AI tools as sources of friction show that, without interpretability and context persistence, fast feedback can create supervisory load rather than reduce it. Second, commits as memory anchors repurpose a feedback mechanism for task resumption, diverging from its version-control function. Third, checklists and short cycles externalize process knowledge through socially-mediated routines. These loops' effectiveness thus depends not only on speed but on the *reliability and persistence* of information—a refinement to the latency-centered view in current DevEx frameworks [47].

Strategies as process improvements rather than accommodations (extends prior framings). Formal accommodations typically require organizational approval and disclosure—processes characterized as bureaucratic or stigma-prone [17, 32]. The strategies here are largely autonomous and embedded in the technical environment, often functioning without disclosure. Within the DevEx perspective, these strategies can be viewed as candidate process improvements that complement individual accommodations. This extends the argument of Newman et al. [33] by shifting the focus from accommodation policies to process design, where practices that support ADHD developers may also improve the experience of colleagues facing similar systemic frictions [7, 35].

5.3 Implications for Research and Practice

For industry practitioners, the findings suggest environmental adjustments that may also benefit the broader developer population: protecting focus time and asynchronous communication to limit interruptions; lightweight process structure (clear definitions of done, short-cycle stories, shared checklists) to aid coordination; and lower feedback latency in CI and code review to ease context reconstruction. Over time, organizations may also reconsider how work practices, role design, and performance evaluation account for different cognitive profiles, treating cognitive diversity as a structural design concern rather than an individual accommodation. Although our data do not compare agile frameworks, they suggest that practices promoting clear goals, visible progress, and short feedback cycles may better support ADHD developers.

For tool and AI design, reliability and persistence appear as relevant as speed. AI assistants could preserve developer context across sessions and clarify the rationale behind their suggestions, so that fast feedback does not become a supervisory burden.

For researchers, the study points to incorporating cognitive diversity into DevEx measurement frameworks. Future work could examine how environments supply the external structure neurodivergent profiles rely on, and test the relational dynamic in Section 5.1, where feedback reliability appears to influence flow sustainability through its effect on cognitive load, with loss, support, and mixed configurations as candidate empirical patterns.

6 Threats to Validity

Following guidance for reflexive thematic analysis [8, 13], we assessed trustworthiness across five criteria.

Credibility. Our purposive sample of ten threads reflects one community; ADHD status is self-reported and unverifiable, and the Score > 400 cutoff favors socially resonant narratives. We mitigated these threats through prolonged engagement, active memoing, coder calibration, and triangulation with prior work on ADHD developers [22, 33].The analysis was conducted by software engineering researchers. Although Authors 1 and 6 have postgraduate training in Autism, ADHD, and Inclusion, none of the researchers has formal clinical training in psychology or psychiatry. Consequently, clinical constructs (e.g., executive functions and hyperfocus) may have been misinterpreted. We grounded interpretation in participants' own language and used multiple rounds of consensus-based coding, but acknowledge the absence of clinical expert validation as a limitation warranting future collaboration with psychology/psychiatry specialists.

Transferability. Findings are bounded by the subreddit's English-speaking, self-selecting audience, the six-month sampling window, and the current salience of AI-assisted development. We support judgments of applicability through thick descriptions, per-thread inclusion rationales, and a versioned keyword lexicon [13, 22, 33].

Dependability. We maintained an audit trail with workflow changes, codebook versioning, and multi-analyst readings; final theme definitions were refined through consensus meetings involving the first two authors.

Confirmability and Ethics. Consistent with reflexive thematic analysis, we prioritized researcher reflexivity over inter-rater reliability [9]. A replication package provides analysis notes and process artifacts [13].

7 Conclusion

This study analyzed how ADHD developers experience the core dimensions of Developer Experience. By identifying specific inhibitors and enablers through the lens of neurodivergence, our findings demonstrate that ADHD-related DevEx is shaped by the tension between personal scaffolding and sociotechnical barriers. While traditional DevEx focuses on operational efficiency, this work reveals that for neurodivergent professionals, these dimensions are deeply tied to attention regulation and memory demands. Ultimately, by showing that the three dimensions of DevEx maintain their structure but shift in content when filtered through executive functions, this study offers a primary path for research and practice to incorporate cognitive diversity without abandoning their established operational vocabulary.

Artifact Availability

A replication package is available [23]. Consistent with Reddit's Terms [37–39], we do not redistribute the raw thread and comment content.

Ethical Considerations on the Use of AI

We declare that Generative Artificial Intelligence tools were used in this work as follows: ChatGPT (OpenAI) assisted with revising English spelling and paragraph clarity during manuscript writing and editing, with documenting the methodology and formatting the Excel file in the supplementary material, and with enhancing Figures 1, 2, and 3. All generated content was reviewed and validated by the authors, who assume full responsibility for the originality and accuracy of the final text, in accordance with CNPq Ordinance No. 2.664/2026.

Acknowledgements

Natalya Goelzer and Gabriel Velloso acknowledge financial support from Dell Brazil through the incentives of the Brazilian Informatics Law (Law No. 8,248/1991). Sabrina Marczak was supported by CNPq (Grant No. 304795/2024-0). Tatiana Cartagena received a graduate scholarship from CAPES. Karina acknowledges that this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

[1] American Psychiatric Association. 2013. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)* (5 ed.). American Psychiatric Publishing, Washington, DC. doi:10.1176/appi.books.9780890425596

[2] Atlassian. 2025. *The State of Developer Experience.* Industry Report CSD-15683. Atlassian. Online survey of 3,500 developers across the US, Australia, France, Germany, the UK, and India (fielded Mar 13–23, 2025) in partnership with Wakefield Research.

[3] Robert D. Austin and Gary P. Pisano. 2017. Neurodiversity as a Competitive Advantage. *Harvard Business Review* 95, 3 (2017), 96–103. https://hbr.org/2017/05/neurodiversity-as-a-competitive-advantage

[4] Sean Ayers-Glassey and Peter D. MacIntyre. 2024. Investigating Emotion Dysregulation and the Perseveration- and Flow-Like Characteristics of ADHD Hyperfocus in Canadian Undergraduate Students. *Psychology of Consciousness: Theory, Research, and Practice* (2024).

[5] Russell A. Barkley, Kevin R. Murphy, and Mariellen Fischer. 2008. *ADHD in Adults: What the Science Says.* The Guilford Press, New York, NY.

[6] Bryce Boe. 2023. *PRAW: The Python Reddit API Wrapper – Documentation, Version 7.7.1.* https://praw.readthedocs.io/en/stable/index.html

[7] Charlotte Brandebusemeyer, Tobias Schimmer, and Bert Arnrich. 2025. Wearables to Measure DevEx at Work. In *IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* 23–33. doi:10.1109/ICSE-SEIP66354.2025.00008

[8] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. doi:10.1191/1478088706qp063oa

[9] Virginia Braun and Victoria Clarke. 2023. Toward Good Practice in Thematic Analysis: Avoiding Common Problems and Becoming a Knowing Researcher. *International Journal of Transgender Health* 24, 1 (2023), 1–6. doi:10.1080/26895269.2022.2129597

[10] Thomas E. Brown. 2008. Executive Functions: Describing Six Aspects of a Complex Syndrome. *Attention Magazine* 15, 1 (2008), 12–17.

[11] David Byrne. 2022. A Worked Example of Braun and Clarke's Approach to Reflexive Thematic Analysis. *Quality and Quantity* 56 (2022), 1391–1412. doi:10.1007/s11135-021-01182-y

[12] Lidiany Cerqueira, Sávio Freire, João Bastos, Rodrigo Spínola, Manoel Mendonça, and José Santos. 2023. A Thematic Synthesis on Empathy in Software Engineering Based on the Practitioners' Perspective. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering (SBES).* Association for Computing Machinery, Campo Grande, Brazil, 332–341. doi:10.1145/3613372.3613407

[13] Daniela S. Cruzes and Tore Dybå. 2011. Recommended Steps for Thematic Synthesis in Software Engineering. In *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM).* IEEE, 275–284. doi:10.1109/ESEM.2011.36

[14] Mihaly Csikszentmihalyi. 1997. *Finding Flow: The Psychology of Engagement with Everyday Life.* Basic Books, New York, NY.

[15] Adele Diamond. 2013. Executive functions. *Annual Review of Psychology* 64 (2013), 135–168. doi:10.1146/annurev-psych-113011-143750

[16] Nicole Doyle. 2020. Neurodiversity at work: a biopsychosocial model and the impact on working adults. *British Medical Bulletin* 135, 1 (2020), 108–125. doi:10.1093/bmb/ldaa021

[17] Daniel Drabble, Emma Cole, Ann Sophie Hahne, Lola Sulola, and Joe Cullen. 2023. *Neurodiversity in the Tech Sector: Global Research on Accessibility, Barriers and How Companies Can Do Better.* Technical Report. Change The Face Alliance.

[18] Alexia Dupuis, Grace Iarocci, and Elizabeth Kelley. 2022. Hyperfocus or Flow? Attentional Strengths in Autism Spectrum Disorder. *Frontiers in Psychiatry* 13 (2022).

[19] Fabian Fagerholm and Jürgen Münch. 2012. Developer Experience: Concept and Definition. In *Proceedings of the International Conference on Software and System Process (ICSSP).* IEEE Press, Zurich, Switzerland, 73–77. doi:10.1109/ICSSP.2012.6225984

[20] Nicole Forsgren, Eirini Kalliamvakou, Abi Noda, Michaela Greiler, Brian Houck, and Margaret-Anne Storey. 2024. DevEx in Action. *Commun. ACM* 67, 6 (May 2024), 42–51. doi:10.1145/3643140

[21] Kiev Gama and Aline Lacerda. 2023. Understanding and Supporting Neurodiverse Software Developers in Agile Teams. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering (SBES).* Association for Computing Machinery, New York, NY, USA, 497–502. doi:10.1145/3613372.3613384

[22] Kiev Gama, Grischa Liebel, Miguel Goulão, Aline Lacerda, and Cristiana Lacerda. 2025. A Socio-Technical Grounded Theory on the Effect of Cognitive Dysfunction in the Performance of Software Developers with ADHD and Autism. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Society.* IEEE Press, Ottawa, Ontario, Canada, 1–12. doi:10.1109/ICSE-SEIS66351.2025.00006

[23] Natalya Marjana Goelzer. 2026. Replication Package for: Understanding ADHD Developers' Challenges Through the Lens of Developer Experience. doi:10.6084/m9.figshare.32162535 Replication package for the SBES 2026 paper.

[24] Michaela Greiler, Margaret-Anne Storey, and Abi Noda. 2023. An Actionable Framework for Understanding and Improving Developer Experience. *IEEE Transactions on Software Engineering* 49, 4 (April 2023), 1411–1425. doi:10.1109/TSE.2022.3175660

[25] Katharina E. Hupfeld, Taylor R. Abagis, and Priti Shah. 2019. Living "in the zone": Hyperfocus in Adult ADHD. *ADHD Attention Deficit and Hyperactivity Disorders* 11 (2019), 191–208. doi:10.1007/s12402-018-0272-y

[26] Nilli Lavie. 2010. Attention, Distraction, and Cognitive Control Under Load. *Current Directions in Psychological Science* 19, 3 (2010), 143–148. doi:10.1177/0963721410370295

[27] Grischa Liebel, Noah Langlois, and Kiev Gama. 2024. Challenges, Strengths, and Strategies of Software Engineers with ADHD: A Case Study. In *IEEE/ACM 46th*

International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). 57–68. doi:10.1145/3639475.3640107

[28] Yimeng Ma, Yu Huang, and Kevin Leach. 2024. Breaking the Flow: A Study of Interruptions During Software Engineering Activities. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE).* Association for Computing Machinery, New York, NY, USA, Article 185, 12 pages. doi:10.1145/3597503.3639079

[29] André N. Meyer, Earl T. Barr, Christian Bird, and Thomas Zimmermann. 2021. Today Was a Good Day: The Daily Life of Software Developers. *IEEE Transactions on Software Engineering* 47, 5 (2021), 863–880. doi:10.1109/TSE.2019.2904957

[30] Giovanni B. Moneta. 2012. On the Measurement and Conceptualization of Flow. In *Advances in Flow Research*, Stefan Engeser (Ed.). Springer, New York, NY, 23–50. doi:10.1007/978-1-4614-2359-1_2

[31] Meredith Ringel Morris, Andrew Begel, and Ben Wiedermann. 2015. Understanding the Challenges Faced by Neurodiverse Software Engineering Employees: Towards a More Inclusive and Productive Technical Workforce. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS).* Association for Computing Machinery, Lisbon, Portugal, 173–184. doi:10.1145/2700648.2809841

[32] Kaia Newman, Sarah Snay, Madeline Endres, Manasvi Parikh, and Andrew Begel. 2025. Disclosure of Neurodivergence in Software Workplaces: A Mixed Methods Study of Forum and Survey Perspectives. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS).* Association for Computing Machinery, Denver, CO, USA, 26 pages. doi:10.1145/3663547.3746334

[33] Kaia Newman, Sarah Snay, Madeline Endres, Manasvi Parikh, and Andrew Begel. 2025. *"Get Me In The Groove": A Mixed Methods Study on Supporting ADHD Professional Programmers.* IEEE Press, 1217–1229. doi:10.1109/ICSE55347.2025.00242

[34] Abi Noda, Margaret-Anne Storey, Nicole Forsgren, and Michaela Greiler. 2023. DevEx: What Actually Drives Productivity? *Commun. ACM* 66, 11 (Oct. 2023), 44–49. doi:10.1145/3610285

[35] Ike Obi, Jenna Butler, Sankeerti Haniyur, Brian Hassan, Margaret-Anne Storey, and Brendan Murphy. 2025. Identifying Factors Contributing to "Bad Days" for Software Developers: A Mixed-Methods Study. In *IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* 1–11. doi:10.1109/ICSE-SEIP66354.2025.00006

[36] Abdul Razzaq, Jim Buckley, Quanjun Lai, Tingting Yu, and Goetz Botterweck. 2024. A Systematic Literature Review on the Influence of Enhanced Developer Experience on Developers' Productivity: Factors, Practices, and Recommendations. *Comput. Surveys* 57, 1, Article 13 (Oct. 2024), 46 pages. doi:10.1145/3687299

[37] Inc. Reddit. [n. d.]. *Data API Terms.* https://redditinc.com/policies/data-api-terms.

[38] Reddit, Inc. 2025. *Public Content Policy.*

[39] Reddit, Inc. 2025. *Reddit Rules.*

[40] Jessica Ribas, Joanne Carneiro, Theo Sousa, Júlia Azevedo, Jailma Januário, Anderson Uchôa, and Juliana Pereira. 2025. Pain in a Safe Space: Mapping Emotions and Discourse in the WomenInTech Subreddit. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software (SBES).* SBC, Porto Alegre, RS, Brasil, 727–733. doi:10.5753/sbes.2025.11171

[41] Judy Robertson, Sandra Gray, Margaret Toye, and Jennifer Booth. 2020. The Relationship Between Executive Functions and Computational Thinking. *International Journal of Computer Science Education in Schools* 3, 4 (2020), 35–49. doi:10.21585/ijcses.v3i4.76

[42] Júlio Silva, Lidiany Cerqueira, Beatriz Santana, Glauco Carneiro, Sávio Freire, Manoel Mendonça, and José Santos. 2025. "Frustrating, Stressful, and Overwhelming": Insights into Software Practitioners' Productivity from Stack Exchange Discussions. In *Proceedings of the 39th Brazilian Symposium on Software Engineering (SBES).* SBC, Porto Alegre, RS, Brasil, 237–248. doi:10.5753/sbes.2025.9914

[43] Edmund J. S. Sonuga-Barke, Panagiota Bitsakou, and Margaret Thompson. 2010. Beyond the Dual Pathway Model: Evidence for the Dissociation of Timing, Inhibitory, and Delay-Related Impairments in Attention-Deficit/Hyperactivity Disorder. *Journal of the American Academy of Child and Adolescent Psychiatry* 49, 4 (2010), 345–355. doi:10.1016/j.jaac.2009.12.018

[44] Sparta Global. 2023. *EqualTech Report 2023: How Neurodiverse Individuals Bring In-Demand Perspectives to Tech.* Technical Report. Sparta Global.

[45] Stack Overflow. 2024. 2024 Stack Overflow Developer Survey. https://survey.stackoverflow.co/.

[46] Margaret-Anne Storey, Thomas Zimmermann, Christian Bird, Jacek Czerwonka, Brendan Murphy, and Eirini Kalliamvakou. 2021. Towards a Theory of Software Developer Job Satisfaction and Perceived Productivity. *IEEE Transactions on Software Engineering* 47, 10 (2021), 2125–2142. doi:10.1109/TSE.2019.2944354

[47] Adeeb Valiulla. 2025. Developer Experience (DevEx) Measurement in the Age of AI. *SSRN Electronic Journal* (June 2025). doi:10.2139/ssrn.5316738