Go with the Flow: Software Engineers and Distractions

Sabine Janssens
StressLabo
Loonbeek, Belgium
sabine@stresslabo.be

Vadim Zaytsev
Formal Methods and Tools
University of Twente
Enschede, The Netherlands
vadim@grammarware.net

ABSTRACT

Ideally, software engineers work in the state of flow: performing challenging tasks like system modelling with a series of routine actions, incorporating immediate IDE feedback, experiencing time distortion and managing harmful interruptions when necessary. We hypothesise that the ability to maintain the state of flow and the skill to get back into flow faster after being interrupted, are essential skills for software engineers, and that developing these increases the capacity to become good software engineers if given sufficient guidance and education. In this position paper, we link flow to trance and contemplate how techniques from the world of sport psychology can be used to teach learners to become better at software design, modelling, programming and debugging, as well as suggesting a means to assess the potential future success of undergraduate study seekers.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; **Socio-technical systems**; • **Software and its engineering** → *Collaboration in software development*; *Software design techniques.*

KEYWORDS

Flow, hypnosis, trance, induction, modelling, programming.

ACM Reference Format:
Sabine Janssens and Vadim Zaytsev. 2022. Go with the Flow: Software Engineers and Distractions. In *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MoDELS '22 Companion), October 23–28, 2022, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3550356.3559101

1 INTRODUCTION

It is widespread anecdotal knowledge that software engineers need to be in the state of flow or "in the zone" in order to be productive. At least to some extent it is also supported by empirical evidence. For instance, software engineering activities like modelling and programming align with the major components of flow identified by Csíkszentmihályi [7], conform to all three factors crucial to consider it to be flow (difficult enough challenges, merged action and awareness and clear goals/feedback) [8]. Some frameworks for analysing flow, like Person–Artefact–Task, also seem directly applicable to flow states that involve human-computer interaction [12]. There is also a known positive correlation between flow and psychological capital and creativity for software developers [51], which is an important quality to gain or keep a competitive edge.

Hermans notes that programmers require a warmup time to get "in the zone" [18, p.184], and attributes it mostly to building a mental model of the code. She proposes some ways to prepare for interruptions, based mostly on improving the working memory. Many observations, like one that experiencing anxiety can disturb one's flow [50], suggest that staying in flow is not exclusively linked to having good memory and a reliable mental model. Since interruptions are known to both increase mental workload and lower performance [10], there is much research conducted on interruption management in software development, focused on interruption value evaluation [14], disruptiveness of self-interruptions [1], coping strategic frameworks [29], etc.

In the context of improving IDE support for flow, Kuusinen et al [27] have done an investigation on 50+ developers, focusing on their user experience (NB: here it is not the user experience of the end user, but the user experience of a software developer as someone using a development environment). They found flow measures to be a good predictor for overall developer experience, but also had many other interesting findings like the perceived freedom to choose a tool positively affects enjoyment and productivity and negatively impacts frustration.

Sheth et al propose to gamify software engineering to improve flow practices by providing achievements like "The Bugslayer" for an engineer that fixes the most bugs in a project or awarding completing work tasks with points that can be used in the company cafeteria or parking [44]. In this paper, we focus much less on extrinsic motivation and more on the inductive way towards the intrinsicly motivated software engineer. For readers interested in gamification in software engineering, we refer to a more recent and much more systematic treatment of the topic by Dal Sasso et al [9].

Caveat lector: the multidisciplinary nature of this paper could occasionally make it hard to read. Please beware that in many cases the same terms mean different things in psychology and computer science, and in this paper we will mostly need the psychological interpretations. Thus, "flow" will not be related to either control flow nor data flow, "visualisation" will not imply any software visualisation, etc.

2 SOFTWARE ENGINEERS AND FLOW

Having established some conceptual connection between flow and software engineering, lets us investigate it deeper and more systematically. According to Csíkszentmihályi, the phenomenology of enjoyment has eight major components [7], which we revisit in detail below:

(1) **A Challenging Activity that Requires Skills** — the easiest to link to software engineering, given the amount of studying and certification expected from accomplished software engineers.
(2) **The Merging of Action and Awareness** — in other words, the lack of necessity to reflect on the result and effect of each action, since each action "carries us forward as if by magic" [7]. Composers can get this effect while jotting down notes of the next musical passage, just as modellers do by adding a meaningful box and connecting it to the rest of the diagram, both by doing so contributing to creating the abstraction of a system. Similarly, both musicians and programmers get closer to their goals by pressing the right buttons on their keyboards.
(3) **Clear Goals and Feedback** — the clarity here does not necessarily mean well-definedness [41] of lack of ambiguity, since flow is well applicable to composers or painters, it is more of "know it when you see it" feeling of accepting or rejecting the final result of your endeavours. This corresponds well to software engineers' knowledge and awareness about software quality. The feedback in the case of software engineering tasks takes a form of syntax highlighting, IDE-embedded smell detection, testing framework results, etc.
(4) **Concentration on the Task at Hand** without worrying about global unrelated issues, is commonly achieved by the sheer complexity of a typical task in software engineering, at least if we follow Csíkszentmihályi's line of reasoning. Our conscious mind is only capable of holding 3–4 distinct items in its visual working memory [40], which occupies most neurotypical brains enough by crafting an abstraction and leaves no space for the mind to wander simultaneously into irrelevant directions. According to more generally applicable attention restoration theory [34], directed attention, also known as voluntary attention, representing the ability to focus on a task on demand, because you decide to focus on that task, is finite, prone to fatigue, and requires planned restorative activities. How much attention fatigue is caused by the effort to ignore a distraction, depends on the autotelic experience and in essence on intrinsic motivation.
(5) **The Paradox of Control** is about being so immersed in exercising control over actions with doubtful outcomes, that you cannot control the content of your actions, and addiction wins over freedom. For sports it is easy to imagine being so immersed in the game that you play for the sake of playing, not for the sake of winning. Similar situations happen in software engineering. Imagine a developer asked by the codebase owner to enforce new internal quality guidelines. After analysing the system with metrics, the developer concludes that its testability [22] suffers due to classes and methods being too long. Then, during the subsequent refactoring or rearchitecting, the developer is focused on keeping the classes in line with abstractions and on seeking ways of splitting large methods into a pipeline of smaller ones — and not necessarily on the concept of maintainability, and definitely not on the thought of keeping the manager happy. This is one of the reasons techniques like pair programming [49] exist: to introduce another role, called observer of navigator, to keep track of the big picture and gently guide the flow of the other developer.
(6) **The Loss of Self-Consciousness** — there is no room for self-scrutiny in flow, since "the self" is not threatened, goals are set, rules are stable and challenges matched by skill levels. If the conditions are optimal, people do not feel the need to focus on the self, protecting the sense of self during flow.
(7) **The Transformation of Time** — spending more time on the task than intended without realising it before the task is completed, is a familiar experience to all software engineers.
(8) **The Autotelic Experience** — the desire to do the thing for its own sake and not for something else. For instance, when a software engineer wants to make a sequence diagram match the class diagram, they do that for the sake of that matching, not overly bothered at that very moment by external factors, even motivating ones like deadlines or payment.

The first three components have been claimed by Csíkszentmihályi et al to be crucial to achieve flow for an immersed individual [8]: clear goals, immediate feedback and a challenge-skills match. The other components are associated with a state of flow, these phenomena can be related to the work that software engineers perform on a daily basis.

3 FLOW IS LIGHT TRANCE

Psychology, just as software engineering, is a large research field with many studies and directions within. Flow is a construct of *positive psychology* which aims at "positive human functioning and flourishing on multiple levels" [43] — hence Csíkszentmihályi's focus on enjoyment and sources of happiness when researching flow. Trance, on the other hand, comes from the study of unconscious mind [11], and finds many practical uses in modern solution-focused hypnotherapy. Most research on trance is focused on intentional and deep trance. These theories are not in conflict, but it is important to be aware that they see the same problem from different angles and thus can provide complementary insights.

Flow is a form of light trance. One does not need to go into deep trance to perform any software engineering activities (more on this in section 4). However, it is known that learning proper trance techniques can improve the performance of people working in flow, such as athletes [35, 45]. Trainings like those used on basketball players [35] or similar to those, can also be used to help computer science students (or other learners of software engineering) with low capacity for entering and staying in flow.

One can think of this as the possibility to work with studying software engineers on one of three levels. First, we can focus on content (this is a class diagram, this is a while statement) and expect them to discover flow on their own, either mentioning it in passing or not at all. Since most of contemporary and precursory computer science education is of that kind, we do not focus on this option in the scope of this paper. The second option is to work with visualisation techniques [46], an approach directly appealing to past accounts of successful flow experiences, in order to re-establish it. Visualisation is a clinical technique that is used since the 1980s to help athletes to sharpen focus and restore confidence in their abilities through a mental rehearsal of their performance [33]. Since visualisation does not remain on the linguistic and cognitive side, it avoids triggering self-critique and other impeding mechanisms, and thus is more effective [16]. The third option and the one we advocate for here, is to introduce new methods in computer science and engineering education that are aligned with modern psychology, such as utilisation and hypnotic inductions[1] [11]. We will elaborate on them over the course of the next sections. To the best of our knowledge, this is the first instance of research explicitly considering trance in the context of software engineering activities.

4 LEARNING FLOW

During deep trance, brain activity shifts from the left analytical to the right experimental mode of thinking. This occurs on a spectrum from flow to deep trance [13]. Deeper trance is related to diminished susceptibility to distractions. People that can more easily get into trance, find it also easier to get into flow [15]. Practising this shift could make software engineers more resilient to distractions and teach them to get back into flow easier. Creating this shift is something that is not discussed in the flow theory.

Many people are following Csíkszentmihályi in assuming that flow "just happens" when they think hard and have a clear problem at hand. This is not universally true, since some people can simply lack good practices to get into flow or to remain in flow. If that is the case, then even learning all the details of existing modelling and programming languages will not make a person less prone to distractions when applying that knowledge. Luckily, this missed opportunity can be repaired relatively easily.

While students with high-functioning autism have comparable mathematical skills to neurotypical students [5], and they have the capacity to focus and even overfocus, it is usually harder for them to properly shift their attention and choose what to focus on. One of the techniques for them is increasing the load until it engages full capacity [39], which typically leads to the opposite results in non-autists. Students with ADHD also have a problem with sustained focus and in general lack the initial attention expected for flow, and need to follow checklists, seek social reminders, limit objects in their field of vision, etc [48]. Dyslexics, when expending effort in order to focus, also tend to overfocus on reading accuracy at the expense of comprehension [38] (e.g., focus so much on names that they overlook associations between classes). In these and many other cases of facing neurodiversity the "just do it" approach of Csíkszentmihályi fails. In neurotypical cases it just puts a natural cap on how far can someone go on discipline alone, because, as we explained in section 2, attention is a limited resource which gets exhausted without proper care.

There have been attempts in the past to measure software engineers mental workload or effort, and provide feedback that way. The work ranged from developing taxonomies of beginner mistakes [3, 28] to working directly with biofeedback from wearable EEG devices [6, 26]. Here we focus on hypnotherapeutic methods.

One of the great starting points in teaching students to get into flow is to reuse past experience of being in flow to initiate another flow experience, and there is one activity that many people have had, and that is associated with flow: gaming. The flow theory is in general one of the best explanations of the state of absorption [50], and playing games has long been known to be related to flow [32]. Addicted gamers have also been investigated in the context of trance [42]: Schimmenti and Caretti discuss how during intense absorption with gaming, people experience an alteration in the state of consciousness, time-distortion, changes in the sense of self. This leads to problematic identification with the virtual world and alternative sensory experiences. However, there is normal non-addictive immersion in many technological tools, and people that get *too* immersed in gaming to escape uncomfortable stressors or memories, can end up in a dissociative trance. Then they lose the normal sense of identity, overidentify with a character or a part of a virtual world, and need psychotherapy to get back into a healthy state. This is not the kind of trance that we describe in this article, but it is important to mention, since it is more readily discussed in the media than the healthy useful states of trance. Without any guidance, the focus of one's attention during trance can go in any direction.

5 INTERVENTION

Consciously trying to link a certain action with a state of flow generally does not lead to the desired results. With untrained people, just wanting to focus when you open your laptop could lead to a lot of frustration when this fails to manifest. Associating a state of relaxed attention, as you get with flow, with a certain action is the easiest when you are in the suggestible state that is natural in trance.

Objects or actions linked to past positive experience of flow, can be used to reenter flow. They can be natural triggers (opening a laptop or putting a hand on a mouse) or unnatural triggers (putting on programming socks or a modelling hat) [30]. The advantage of using natural triggers is that they, once established, require no conscious effort.

In sports, flow can be taught by procedures that install a trigger via hypnotic imagery. With a hypnotic induction participants are invited into a trance state, hypnotic regression to a past success is evoked and an appropriate trigger suggested [36]. The intervention we propose, comprises the following steps, based on a procedure applied to athletes following this induction-regression-trigger procedure [30, 36]:

(1) Participants are invited to sit comfortably and breathe deeply, while counting backwards from 10 to 1.
(2) The participants get a 15 minutes progressive muscle relaxation session. This involves relaxing parts of the body in a systematic way and contrasting the feeling of tension with a feeling of relaxation in the muscles.
(3) Go down the imaginary stairs, one step at a time for 20 steps. Ericksonian suggestions (e.g., "you can feel more relaxed with every step") help to bring participants into a state of

[1]Using these techniques imply obtaining informed consent: students must get a short text explaining the procedure and have the possibility to opt out at any moment.

trance. People are invited to notice things they can feel while going down a flight of stairs.
(4) Participants are told they can find a door at the bottom of the staircase with a comfortable chair in the adjacent room. They are invited to sit on that chair and focus on a small cinema screen with a relaxing scene on it (e.g., a flowery meadow).
(5) Participants are asked to direct their attention to the pleasant and relaxing scenes of their choice.
(6) Participants are asked to remember one of their best flow experiences with as many sensory details as possible (e.g., where they were sitting, what were they wearing, how cold or warm it was, what they could hear). In this step people are as immersed in the memory as possible, suppressing other non-relevant memories [36];
(7) A trigger gets introduced to link this sensory experience of flow for easier recall outside of the trance state (e.g., their mouse).
(8) Participants are told to get out of the chair, pass the door and ascend the stairs. They are told they can get out of trance, feeling alert and refreshed.

Hypnosis has been safely used for decades by trained hypnotherapists in clinical and experimental settings [4, 20, 21]. It is known to be effective and safe on all neurodiverse subjects except heavy dementia patients [47] (for which adaptations are required) and people in the middle of a psychotic episode [19] (for which a therapeutic alliance must be formed) — both unlikely to be encountered among students. The extent to which people can benefit from hypnotic interventions largely depends on their baseline suggestibility, which tends to be a relatively stable difference between individuals [31, 37]. Step 3 is based on the Edmund Jacobson progressive muscle relaxation technique proposed almost a full century ago [25] but still used actively in treating stressed patients. Other steps borrow from Milton Erickson [11] or other clinicians that use hypnosis to improve performance and confidence. Any experienced hypnotherapist usually bases suggestions on scripts from a reference book and alters them to fit their style. In this way a hypnotic intervention from one study will never be completely the same as another (except if they use the same recordings). The use of intonation and stresses on the right words is essential for the right effect. For this procedure it is therefore essential to have an experienced hypnotherapist that can adjust this protocol to fit the target audience and specific situation. After the initial, live intervention, participants are required to listen to a recording for the subsequent week to strengthen the connection between the trigger and the desired flow experience [17].

6 METHOD

Each phase of measurements should cover two parts: checking how deep students are in trance and how good do they perform as software engineers. In absence or the impossibility to make use of biofeedback devices, the depth of flow can be measured with questionnaires like FSS [23] or its improved variants like SDFS-2 [24] — this Flow State Scale is being used to measure flow during sports and physical activity. With this self-report scale based on the dimensions of flow discussed by Csíkszentmihályi, a previous flow experience can be measured [23]. In sports psychology, mental skills are habitually taught to athletes to improve their performance [30] with a variety of techniques. In the context of software engineering, performance is traditionally measured as a counter or a degree of success in completing (re)engineering tasks, possibly complemented by some assessment of the internal quality of the result (e.g., with a defect density or a feature coverage metric).

Throughout this paper we were using the term "student" to refer to a software engineer on some point of the learning curve where significant observable process can still be made. These can be literally university students, school pupils, junior developers, etc. In any case we would need to separate the available population into two groups: the testing group will do the intervention as described in section 5, and the control group will only get the first step of it with the advice to listen to their favourite music for the same duration as the recording of the other group, instead of the direct instructions and suggestions. Both groups will be checked on following these instructions and their experience. Quantitatively comparing the results obtained separately from the testing group and the control group, will help to take into account any general progress made independently from this training.

We envision one instance of this double measurement to take place before the intervention is even introduced, to obtain a baseline; then once after several days of practising the intervention steps, and then one final time after 7+ more days of interventions.

7 CONCLUDING REMARKS

In this paper we treated flow [2, 7, 8, 12, 23, 24, 27, 30, 32, 35, 45, 50, 51] as a form of light trance [11, 25, 42] in the context of software engineering. We found ways to associate the skills of initiating flow and maintaining flow to developer performance. Combined with the association to developer enjoyment, they form then an attractive target to use in education. We have proposed a concrete intervention (section 5) and a concrete experimental setup (section 6) to be used in the future. The next step is to use them in a pilot study and then on a larger scale experiment, and only then move on to draw conclusions.

After collecting information from the proposed experiments, we plan to investigate deeper whether different activities within software engineering have their own unique relation to flow and trance. If this is so, then possibly techniques that work to teach students requirements elicitation will be different from the techniques needed for modelling the system architecture, different yet from techniques for programming and for performing reactive corrective maintenance activities. All these activities are related to flow, as we argued in section 2, but it is sensible to assume that some connect better to the autotelic experience, being more rewarding; some others pertain to the immediacy of feedback, being well-supported by tooling, etc.

Given the pioneering nature of this planned experiment, we have focused primarily on the depth of flow in an individual software developer. In a far future, further investigations can be designed and performed on the level of pairs of programmers, then on the level of local teams, and finally on global software engineering teams.

REFERENCES

[1] Zahra Shakeri Hossein Abad, Oliver Karras, Kurt Schneider, Ken Barker, and Mike Bauer. 2018. Task Interruption in Software Development Projects: What Makes Some Interruptions More Disruptive than Others?. In *EASE*. ACM, 122–132. https://doi.org/10.1145/3210459.3210471

[2] Kiyoshi Asakawa. 2004. Flow Experience and Autotelic Personality in Japanese College Students: How do they Experience Challenges in Daily Life? *JoHS* 5, 2 (2004), 123–154. https://doi.org/10.1023/B:JOHS.0000035915.97836.89

[3] Piraye Bayman and Richard E Mayer. 1983. A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements. *Commun. ACM* 26, 9 (1983), 677–679. https://doi.org/10.1145/358172.358408

[4] Jared W. Bollinger. 2018. The Rate of Adverse Events Related to Hypnosis During Clinical Trials. *American Journal of Clinical Hypnosis* 60, 4 (2018), 357–366. https://doi.org/10.1080/00029157.2017.1315927

[5] Hsu-Min Chiang and Yueh-Hsien Lin. 2007. Mathematical Ability of Students with Asperger Syndrome and High-Functioning Autism: A Review of Literature. *Autism* 11, 6 (2007), 547–556. https://doi.org/10.1177/1362361307083259

[6] Ricardo Couceiro, Gonçalo Duarte, João Durães, João Castelhano, Catarina Duarte, César A. D. Teixeira, Miguel Castelo-Branco, Paulo Carvalho, and Henrique Madeira. 2019. Pupillography as Indicator of Programmers' Mental Effort and Cognitive Overload. In *DSN*. IEEE, 638–644. https://doi.org/10.1109/DSN.2019.00069

[7] Mihály Róbert Csíkszentmihályi. 1990. *Flow: the Psychology of Optimal Experience.* Harper and Row.

[8] Mihály Róbert Csíkszentmihályi, S. Abuhamdeh, and J. Nakamura. 2005. Flow. In *Handbook of Competence and Motivation.* Guilford Press, 598–698.

[9] Tommaso Dal Sasso, Andrea Mocci, Michele Lanza, and Ebrisa Mastrodicasa. 2017. How to Gamify Software Engineering. In *SANER*. IEEE, 261–271. https://doi.org/10.1109/SANER.2017.7884627

[10] Elmira Zahmat Doost and Wei Zhang. 2022. Mental Workload Variations during Different Cognitive Office Tasks with Social Media Interruptions. *Ergonomics* 0, 0 (2022), 1–17. https://doi.org/10.1080/00140139.2022.2104381

[11] Milton H. Erickson. 1958. Naturalistic Techniques of Hypnosis. *American Journal of Clinical Hypnosis* 1, 1 (1958), 3–8. https://doi.org/10.1080/00029157.1958.10401766

[12] Christina M Finneran and Ping Zhang. 2003. A Person–Artefact–Task (PAT) Model of Flow Antecedents in Computer-Mediated Environments. *International Journal of Human-Computer Studies* 59, 4 (2003), 475–496. https://doi.org/10.1016/S1071-5819(03)00112-5

[13] Pierre Flor-Henry, Yakov Shapiro, and Corine Sombrun. 2017. Brain Changes During a Shamanic Trance: Altered Modes of Consciousness, Hemispheric Laterality, and Systemic Psychobiology. *Cogent Psychology* 4, 1 (2017), 25. https://doi.org/10.1080/23311908.2017.1313522

[14] Sukeshini Grandhi and Quentin Jones. 2010. Technology-Mediated Interruption Management. *International Journal of Human-Computer Studies* 68, 5 (2010), 288–306. https://doi.org/10.1016/j.ijhcs.2009.12.005

[15] J. Robert Grove and Michael A. E. Lewis. 1996. Hypnotic Susceptibility and the Attainment of Flowlike States during Exercise. *Journal of Sport and Exercise Psychology* 18, 4 (1996), 380–391. https://doi.org/10.1123/jsep.18.4.380

[16] Eric Hall, Carol Hall, Pamela Stradling, and Diane Young. 2006. *Guided Imagery: Creative Interventions in Counselling & Psychotherapy.* Sage.

[17] D. Corydon Hammond (Ed.). 1990. *Handbook of Hypnotic Suggestions and Metaphors.* W.W. Norton & Company.

[18] Felienne Hermans. 2021. *The Programmer's Brain: What Every Programmer Needs to Know about Cognition.* Manning.

[19] James R. Hodge. 1988. Can Hypnosis Help Psychosis? *American Journal of Clinical Hypnosis* 30, 4 (1988), 248–256. https://doi.org/10.1080/00029157.1988.10402747

[20] Winfried Häuser, Maria Hagl, Albrecht Schmierer, and Ernil Hansen. 2016. The Efficacy, Safety and Applications of Medical Hypnosis: A Systematic Review of Meta-analyses. *Deutsches Ärzteblatt International* 113, 17 (2016), 289–296. https://doi.org/10.3238/arztebl.2016.0289

[21] Kenneth V. Iserson. 2014. An Hypnotic Suggestion: Review of Hypnosis for Clinical Emergency Care. *The Journal of Emergency Medicine* 46, 4 (2014), 588–596. https://doi.org/10.1016/j.jemermed.2013.09.024

[22] ISO/IEC 25010. 2011. ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[23] Susan A. Jackson and Herbert W. Marsh. 1996. Development and Validation of a Scale to Measure Optimal Experience: The Flow State Scale. *JSEP* 18, 1 (1996), 17–35. https://doi.org/10.1123/jsep.18.1.17

[24] Susan A. Jackson, Andrew J. Martin, and Robert C. Eklund. 2008. Long and Short Measures of Flow: The Construct Validity of the FSS-2, DFS-2, and New Brief Counterparts. *JSEP* 30, 5 (2008), 561–587. https://doi.org/10.1123/jsep.30.5.561

[25] Edmund Jacobson. 1924. The Technic of Progressive Relaxation. *The Journal of Nervous and Mental Disease* 60, 6 (1924), 568–578.

[26] Makrina Viola Kosti, Kostas Georgiadis, Dimitrios A. Adamos, Nikos Laskaris, Diomidis Spinellis, and Lefteris Angelis. 2018. Towards an Affordable Brain Computer Interface for the Assessment of Programmers' Mental Workload. *IJHCS* 115 (2018), 52–66. https://doi.org/10.1016/j.ijhcs.2018.03.002

[27] Kati Kuusinen, Helen Petrie, Fabian Fagerholm, and Tommi Mikkonen. 2016. Flow, Intrinsic Motivation, and Developer Experience in Software Engineering. In *XP*. Springer, 104–117. https://doi.org/10.1007/978-3-319-33515-5_9

[28] Ralf Lämmel, Simon J. Thompson, and Markus Kaiser. 2013. Programming Errors in Traversal Programs over Structured Data. *SCP* 78, 10 (2013), 1770–1808. https://doi.org/10.1016/j.scico.2011.11.006

[29] Jay Liebowitz. 2011. Interruption Management: A Review and Implications for IT Professionals. *IT Professional* 13, 2 (2011), 44–48. https://doi.org/10.1109/MITP.2010.118

[30] Pete Lindsay, Ian Maynard, and Owen Thomas. 2005. Effects of Hypnosis on Flow States and Cycling Performance. *The Sport Psychologist* 19, 2 (2005), 164–177. https://doi.org/10.1123/tsp.19.2.164

[31] Eric C. Meyer and Steven Jay Lynn. 2011. Responding to Hypnotic and Nonhypnotic Suggestions: Performance Standards, Imaginative Suggestibility, and Response Expectancies. *IJCEH* 59, 3 (2011), 327–349. https://doi.org/10.1080/00207144.2011.570660

[32] Fiona Fui-Hoon Nah, Brenda Eschenbrenner, Qing Zeng, Venkata Rajasekhar Telaprolu, and Sepandar Sepehr. 2014. Flow in Gaming: Literature Synthesis and Framework Development. *International Journal of Information Systems and Management* 1, 1-2 (2014), 83–124. https://doi.org/10.1504/IJISAM.2014.062288

[33] Thomas Newmark. 2012. Cases in Visualization for Improved Athletic Performance. *Psychiatric Annals* 42, 10 (2012), 385–387. https://doi.org/10.3928/00485713-20121003-07

[34] Heather Ohly, Mathew P. White, Benedict W. Wheeler, Alison Bethel, Obioha C. Ukoumunne, Vasilis Nikolaou, and Ruth Garside. 2016. ART: A Systematic Review of the Attention Restoration Potential of Exposure to Natural Environments. *JTEH* 19, 7 (2016), 305–343. https://doi.org/10.1080/10937404.2016.1196155

[35] John Pates, Andy Cummings, and Ian Maynard. 2002. The Effects of Hypnosis on Flow States and Three-Point Shooting Performance in Basketball Players. *The Sport Psychologist* 16, 1 (2002), 34 – 47. https://doi.org/10.1123/tsp.16.1.34

[36] John Pates and Ian Maynard. 2000. Effects of Hypnosis on Flow States and Golf Performance. *Perceptual and Motor Skills* 91, 3.2 (2000), 1057–1075.

[37] Carlo Piccione, Ernest R Hilgard, and Philip G Zimbardo. 1989. On the Degree of Stability of Measured Hypnotizability over a 25-year Period. *Journal of Personality and Social Psychology* 56, 2 (1989), 289.

[38] Gavin Reid and Jennie Guise. 2019. *Assessment for Dyslexia and Learning Differences: A Concise Guide for Teachers and Parents.* Jessica Kingsley Publishers.

[39] Anna M. Remington, John G. Swettenham, and Nilli Lavie. 2012. Lightening the Load: Perceptual Load Impairs Visual Detection in Typical Adults but not in Autism. *Journal of Abnormal Psychology* 121, 2 (2012), 544. https://doi.org/10.1037/a0027670

[40] Jeffrey N. Rouder, Richard D. Morey, Nelson Cowan, Christopher E. Zwilling, Candice C. Morey, and Michael S. Pratte. 2008. An Assessment of Fixed-Capacity Models of Visual Working Memory. *Proceedings of the National Academy of Sciences* 105, 16 (2008), 5975–5979. https://doi.org/10.1073/pnas.0711295105

[41] Arthur Rump and Vadim Zaytsev. 2022. A Refined Model of Ill-definedness in Project-Based Learning. In *EduSymp*. https://doi.org/10.1145/3550356.3556505

[42] Adriano Schimmenti and Vincenzo Caretti. 2017. Video-terminal Dissociative Trance: Toward a Psychodynamic Understanding of Problematic Internet Use. *Clinical Neuropsychiatry* 14, 1 (2017), 64–72.

[43] Martin Elias Peter Seligman and Mihály Róbert Csíkszentmihályi. 2014. Positive Psychology: An Introduction. In *Flow and the Foundations of Positive Psychology.* Springer, 279–298.

[44] Swapneel Sheth, Jonathan Bell, and Gail Kaiser. 2011. HALO (Highly Addictive, Socially Optimized) Software Engineering. In *GAS*. ACM, 29–32. https://doi.org/10.1145/1984674.1984685

[45] Scott Sinnett, Joshua Jäger, Sarah Morgana Singer, and Roberta Antonini Philippe. 2020. Flow States and Associated Changes in Spatial and Temporal Processing. *Frontiers in Psychology* 11 (2020), 381. https://doi.org/10.3389/fpsyg.2020.00381

[46] Valerie Thomas. 2015. *Using Mental Imagery in Counselling and Psychotherapy: A Guide to More Inclusive Theory and Practice.* Routledge.

[47] Emilie Wawrziczny, Amandine Buquet, and Sandrine Picard. 2021. Use of Hypnosis in the Field of Dementia: A Scoping Review. *Archives of Gerontology and Geriatrics* 96 (2021), 104453. https://doi.org/10.1016/j.archger.2021.104453

[48] Tom Whiteman, Michele Novotni, and Randy Petersen. 1995. *Adult ADD: A Reader Friendly Guide to Identifying, Understanding, and Treating Adult Attention Deficit Disorder.* Pinon Press.

[49] Laurie Williams. 2001. Integrating Pair Programming into a Software Development Process. In *CSEE*. IEEE, 27–36. https://doi.org/10.1109/CSEE.2001.913816

[50] Kazuki Yoshida, Kiyoshi Asakawa, Taro Yamauchi, Satoshi Sakuraba, Daisuke Sawamura, Yui Murakami, and Shinya Sakai. 2013. The Flow State Scale for Occupational Tasks: Development, Reliability, and Validity. *HKJOT* 23, 2 (2013), 54–61. https://doi.org/10.1016/j.hkjot.2013.09.002

[51] Aisha Zubair and Anila Kamal. 2015. Work Related Flow, Psychological Capital, and Creativity Among Employees of Software Houses. *Psychological Studies* 60, 3 (2015), 321–331. https://doi.org/10.1007/s12646-015-0330-x