

DIA 2

Je zal het wel herkennen: je zit vast in een nieuwe functionaliteit. Ondanks alle kennis die je hebt lijkt niks te werken. Uit frustratie besluit je maar even je computer te verlaten, of dit nou een bakje koffie is of een wandeling of een andere vorm van ontspanning. Voor mij is dit bij uitstek een bak koffie bij mijn favoriete koffiezaak van een bevriende ondernemer: met regelmaat krijg ik daar een eureka moment en begin ik tegen hem te ratelen over de oplossing.

DIA 3

Grinnikend kijken we elkaar dan aan... Want hij heeft meestal geen idee wat de wonderbaarlijke openbaring is.

De puzzelstukjes vallen op zijn plaats; op zo'n moment is het niet zozeer de code die je tegenhoudt maar ook de manier waarop jouw brein ermee omgaat. Code schrijven is niet alleen een technische vaardigheid: het is ook een mentale kunst.

Vandaag gaan we dieper in op hoe ons brein werkt tijdens programmeren en hoe we daar optimaal gebruik van kunnen maken.

Voorstellen

Goedenavond, mijn naam is Haimanti Dekker! Welkom iedereen, ontzettend tof dat er zo'n grote groep mensen voor mij staat.

Ik ben een PHP developer en werk op dit moment voornamelijk met Magento. Voor side projects of applicaties tegen Magento aan maak ik gebruik van Symfony of Laravel.

Ondertussen schrijf ik ongeveer 14 jaar code. Van front-end, tot backend en tegenwoordig fullstack.

Door de jaren heen heb ik niet alleen kennis vergaard over code schrijven maar ook over het brein.

Die interesse is er eigenlijk altijd wel geweest; op jonge leeftijd was ik mij al bewust van het feit dat mijn brein soms wat sneller ging dan eigenlijk handig was. Dus begon ik met lezen: hoe werkt focus en wat het is; doet mijn brein echt dingen anders en kan ik dat beïnvloeden? De uiteindelijke vraag: hoe kan ik mijn brein voor mij laten werken ipv tegen mij.

Deel 1

1. Anatomie van het brein

Je brein is een orgaan dat bestaat uit miljarden neuronen. Deze neuronen communiceren met elkaar. Dit netwerk van neuronen maakt het mogelijk om gedachten, emoties en bewegingen te genereren.

Neurale netwerken en synaptische verbindingen

Goed, die neuronen zijn verbonden door synapsen, waar de communicatie tussen cellen plaatsvindt. Wanneer we iets nieuws leren of een vaardigheid oefenen, worden deze verbindingen versterkt.

Dit proces, bekend als synaptische plasticiteit, is cruciaal voor leren en geheugen.

De prefrontale cortex

De prefrontale cortex is het deel van het brein dat verantwoordelijk is voor hogere cognitieve functies zoals besluitvorming, probleemoplossing en planning.

Als developers gebruiken we deze hersenregio intensief om logische structuren te creëren en fouten in de code te vinden.

Hoe het brein informatie verwerkt en onthoudt

Ons brein verwerkt informatie door het in verschillende regio's te verdelen, afhankelijk van het type informatie. Visuele informatie wordt bijvoorbeeld anders verwerkt dan taal. Uiteindelijk zijn herhaling en actieve betrokkenheid essentieel voor het opslaan van informatie in het langetermijngeheugen.

Denk aan fietsen. Verleer je niet, toch? Het zit opgeslagen in je langetermijngeheugen (en door het regelmatig te blijven doen - of in mijn geval eerder een keer per maand) - blijf je er ervaren in. Trouwens, fietsen kan bijna niet. Als de hersenverbindingen die dit regelen eenmaal zijn aangemaakt, zijn ze niet meer te verbreken. (muscle memory).

2. Dopamine

Goed, je weet nu iets van brein (althans de anatomie), laten we eens kijken naar dopamine.

Dopamine is een neurotransmitter, een chemische stof die signalen tussen neuronen in de hersenen overbrengt. Het speelt een cruciale rol in veel hersenfuncties, waaronder motivatie, beloning en genot.

Rol van dopamine bij motivatie en beloning

Dopamine wordt (vaak) vrijgegeven als reactie op behaalde doelen of positieve ervaringen. Het creëert een gevoel van beloning, wat ons motiveert om bepaald gedrag te herhalen. Dit is essentieel voor leren en het ontwikkelen van nieuwe vaardigheden.

Invloed van dopamine op programmeren

Bij programmeren kan dopamine een grote rol spelen. Elke keer dat we een probleem oplossen of een fout in de code vinden, ervaren we een kleine golf van dopamine. Dit geeft ons een gevoel van voldoening en stimuleert ons om door te gaan, zelfs bij uitdagende taken.

Door bewust te werken aan kleine, haalbare doelen, kunnen we de dopamineafgifte stimuleren. Dit verhoogt onze motivatie en productiviteit. Bijvoorbeeld, het opsplitsen van een complex project in kleinere taken en elke voltooiing als een succes te vieren, kan ons helpen gefocust en gemotiveerd te blijven. Dus, hup: epics, stories en subtaken. Ze dienen ergens voor!

Gevaren van dopamineverslaving

Hoewel dopamine belangrijk is, moeten we oppassen voor overmatige afhankelijkheid. Activiteiten zoals sociale media en games kunnen leiden tot dopamine verslaving, waarbij ons brein steeds meer stimulatie nodig heeft om dezelfde beloning te ervaren. Dit kan onze focus en productiviteit negatief beïnvloeden.

Door bewust om te gaan met dopamine, kunnen we niet alleen onze motivatie en productiviteit verhogen, maar ook een gezonde balans bewaren in onze dagelijkse activiteiten.

3. Dopamine verslaving, de maatschappij en jij

Goed, last but not least. Laten we ook even de keerzijde belichten van dopamine, moderne technologie en de maatschappij.

De verlaagde aandachtsspanne

In onze moderne wereld worden we voortdurend gebombardeerd met informatie en afleidingen. Zaken zoals smartphones en sociale media hebben ons leven verrijkt, maar ze hebben ook een aanzienlijke invloed op onze aandachtsspanne.

Sociale media en smartphones zijn ontworpen om onze aandacht vast te houden. Constante meldingen en updates trekken voortdurend onze aandacht en zorgen voor een gefragmenteerde concentratie. Dit maakt het (over het algemeen) moeilijker om lange periodes van focus te handhaven.

Gevolgen van een gefragmenteerde aandacht op lange termijn

Op lange termijn kan een gefragmenteerde aandacht leiden tot verminderde productiviteit, verhoogde stressniveaus, en een verminderd vermogen om complexe problemen op te lossen. Het brein raakt gewend aan korte aandachtsspanne, waardoor het moeilijker wordt zogeheten “deep work” te doen en creatieve oplossingen te vinden.

Deel 2 - Je geheugen

1. Lange termijn geheugen

Patronen en herkenning

Ons brein is bedreven in het herkennen van patronen. Dit is een essentieel vermogen voor developers, omdat we voortdurend op zoek zijn naar bekende structuren en logica in code. Patronen helpen ons snel te begrijpen wat een stuk code doet zonder elke regel minutieus te hoeven lezen.

Bijvoorbeeld, als we vaak voorkomende constructies zoals loops, conditionele logica en functies zien, kunnen we snel het doel en de werking afleiden.

Visualisatie van code en het mentale model

Wanneer we code lezen, creëren we mentale modellen die ons helpen om de logica en structuur te begrijpen. Deze mentale modellen zijn visuele en conceptuele representaties van de code in ons hoofd.

Het lange termijngeheugen en developers

Let op; voor ervaren developers interessant en voor als je misschien net start of nog niet zo lang code schrijft. Wat maakt een ervaren developer kundig?

Ervaren developers hebben een breed scala aan patronen en mentale modellen opgebouwd door de jaren heen. Dit is te trainen maar het valt en staat eigenlijk met ervaring door te doen. Vlieguren maken. Dit stelt hen in staat om complexe code sneller te interpreteren en te begrijpen. Wat zijn dan kenmerken die gerelateerd zijn aan ervaring en je brein?

Grote bibliotheek van patronen: Door ervaring hebben ze een uitgebreide verzameling code patronen en bijvoorbeeld best practices opgeslagen in hun geheugen. Dit maakt het gemakkelijker om bijvoorbeeld nieuwe features te schrijven.

Effectieve visualisatie technieken

Door de ruime ervaring kan iemand makkelijker code visualiseren en modelleren in zijn of haar hoofd. Denk aan een complexe module, een developer met meer ervaring zal sneller verbanden en relaties zien tussen logica.

Sterke probleemoplossende vaardigheden

Sneller identificeren van de kern van een probleem doordat ze sneller code kunnen isoleren en analyseren.

Automatisering van routine taken:

Veel van de routinematige cognitieve taken, zoals het herkennen van syntactische structuren en het volgen van eenvoudige logica, zijn geautomatiseerd in het brein van een ervaren developer. Dit laat meer cognitieve ruimte over voor het aanpakken van moeilijke concepten en bugs.

2. Kortetermijngeheugen

Rol van het kortetermijngeheugen in het schrijven en debuggen van code

Het kortetermijngeheugen speelt een cruciale rol bij alledaags werk. Wanneer we code schrijven of debuggen, moeten we vaak meerdere stukken informatie tegelijk in gedachten houden. Dit kunnen variabelen, structuren, of de context.

Het kortetermijngeheugen helpt ons om deze informatie actief te beheren, zodat we effectieve beslissingen kunnen nemen en snel bugs kunnen identificeren.

Ofwel het kortetermijngeheugen gebruiken we veel voor ons dagelijkse werk. Voor meer ervaren developers zie je dat het langetermijngeheugen een belangrijke rol speelt.

Deel 3 - hack je brein

1. Verbeter je kortetermijngeheugen en programmeer efficiënter

Er zijn verschillende technieken die je kunt gebruiken om je kortetermijngeheugen te verbeteren en efficiënter te programmeren.

Chunking

Code voorbeeld

Definitie: Chunking is het proces waarbij je grote hoeveelheden informatie opdeelt in kleinere, beter beheersbare eenheden of "chunks."

Toepassing: In plaats van een hele functie in één keer te proberen te begrijpen, kun je deze opdelen in kleinere logische delen. Dit maakt het gemakkelijker om de werking van elk deel te onthouden en te analyseren.

Herhaling (nieuwe php versie vb) ->

json_validate() -> val je terug op de bekende manier van werken of niet?

Herhaling helpt bij het opslaan van informatie in het kortetermijngeheugen door het meerdere keren te herhalen.

Toepassing: Als je een nieuw concept hebt of leert, herhaal deze dan meerdere keren in verschillende contexten. Dit versterkt de neurale verbindingen en helpt de informatie beter te onthouden.

Mind Mapping (vb)

Definitie: Mind mapping is een visuele techniek die helpt om relaties tussen verschillende stukken informatie te zien en te onthouden.

Toepassing: Maak een mindmap van de codebase, waarbij je de belangrijkste functies en hun onderlinge relaties visueel weergeeft. Dit kan je helpen om de structuur van de code beter te begrijpen en te onthouden.

2. Focuswerk

Ter voorbereiding voor deze sessie pakte ik "Deep work" van Cal Newport nog eens uit de kast.

Het verwijst naar periodes van ononderbroken concentratie waarbij je je volledig richt op een taak. Voor developers is dit cruciaal omdat het helpt bij het effectief oplossen van problemen en het schrijven van kwalitatieve code.

Voordelen van diepe concentratie

Diepe concentratie biedt meerdere voordelen, waaronder:

- Verhoogde productiviteit en efficiëntie
- Betere probleemoplossende vaardigheden: Diepe concentratie stelt je in staat om complexe problemen grondig te analyseren en creatieve oplossingen te vinden.
- Technieken om focus werk te bevorderen
- Er zijn verschillende technieken die je kunnen helpen om focus werk te bevorderen:

Pomodoro techniek

Deze techniek houdt in dat je werkt in blokken van bijv 25 minuten, gevolgd door een korte pauze. Dit helpt je om gefocust te blijven en voorkomt mentale vermoeidheid. * pomodoro, 25m aan focus vereist oefening. Door de korte aandachtsspanne kan dat in het begin lastig zijn. Anderzijds bij grote features kan 25 (waar je echt in een focus flow zit) kan het juist verstoren.

Time blocking

Plan specifieke tijdsblokken in je agenda voor verschillende taken. Dit zorgt ervoor dat je ongestoord de tijd hebt om je op één taak te concentreren.

—

Flow states

Probeer een 'flow' staat te bereiken, waarin je volledig opgaat in je werk. Dit kan door een rustige werkomgeving te creëren en afleidingen te minimaliseren.

Door focus werk in je dagelijkse routine te integreren, kun je als developers je efficiëntie en probleemoplossende vaardigheden aanzienlijk verbeteren.

3. Van brein naar code

Hoe ideeën en concepten omzetten naar werkende code

Het omzetten van ideeën en concepten naar werkende code is een kernvaardigheid voor developers. Dit proces begint met een abstract idee in je brein en eindigt met een functionaliteit. Het vereist zowel creativiteit als technische vaardigheid om deze kloof te overbruggen.

-

Het belang van een goed mentaal model

Een goed mentaal model is cruciaal bij het vertalen van ideeën naar code. Dit model helpt je om de structuur en logica van je oplossing te visualiseren en te begrijpen hoe de verschillende componenten samenwerken.

Je kunt een mentaal model ontwikkelen door:

Actief leren en oefenen: Regelmatig oefenen met verschillende programmeertaken en problemen helpt je om je mentale modellen te verfijnen.

Reflecteren op je code: Neem de tijd om na te denken over de structuur en logica van je code, en hoe deze aansluit bij je oorspronkelijke concept.

Gebruik maken van visuele hulpmiddelen: Tools zoals diagrammen en mindmaps kunnen je helpen om complexe ideeën visueel weer te geven en beter te begrijpen.

Oefeningen en best practices om de kloof tussen denken en coderen te overbruggen

Er zijn verschillende oefeningen en best practices die je kunnen helpen om je ideeën effectief om te zetten in code:

Pseudocode schrijven: Begin met het schrijven van je idee in pseudocode. Dit helpt je om de logica van je oplossing te structureren zonder je te concentreren op syntactische details.

Modulair denken: Verdeel je oplossing in kleinere, onafhankelijke modules of functies. Dit maakt het gemakkelijker om je code te beheren en te debuggen.

Code reviews: Deel je code met collega's voor feedback. Dit kan je helpen om blinde vlekken te identificeren en nieuwe perspectieven op je oplossing te krijgen.

Regelmatige reflectie: Na het voltooien van een project, neem de tijd om te reflecteren op wat goed ging en wat beter kon. Dit helpt je om je vaardigheden continu te verbeteren.

Slot en vragen