

University of Groningen

Computer programming skills: A cognitive perspective

Graafsma, Irene

DOI:
[10.33612/diss.168003240](https://doi.org/10.33612/diss.168003240)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2021

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Graafsma, I. (2021). *Computer programming skills: A cognitive perspective*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen. <https://doi.org/10.33612/diss.168003240>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Samenvatting

Samenvatting

Dit proefschrift onderzoekt het computerprogrammeren vanuit een cognitieve invalshoek. Programmeren is een relatief nieuwe vaardigheid die de afgelopen decennia steeds belangrijker is geworden in onze samenleving. Om deze vaardigheid optimaal te kunnen doceren is het belangrijk om de cognitieve aard van het programmeren te begrijpen. Het is dan ook belangrijk om te onderzoeken of en hoe deze vaardigheid verband houdt met andere cognitieve vaardigheden, persoonlijkheidskenmerken en verwerking in de hersenen. Dit proefschrift behandelt alle drie deze gebieden.

Hoofdstuk 1 geeft een overzicht van de geschiedenis van programmeeronderzoek. Het behandelt de belangrijkste bevindingen in de drie aandachtsgebieden en bespreekt methodologieën die vaak worden gebruikt om programmeervaardigheden te onderzoeken. Vervolgens worden de belangrijkste onderzoeksvragen van het proefschrift geformuleerd en worden de onderwerpen van de volgende hoofdstukken geïntroduceerd.

Hoofdstuk 2 van het proefschrift legt de basis voor de Hoofdstukken 3 en 4 door twee verkorte versies van de SCS1 (een programmeertest voor beginnende informaticastudenten) te valideren. De twee verkorte versies bleken niet als volledig gelijk te kunnen worden beschouwd. Daarnaast was één versie van twijfelachtige kwaliteit. De andere versie had echter een vergelijkbare betrouwbaarheid en validiteit als de originele, volledige test, die in eerdere onderzoeken was gevalideerd. Deze studie resulteert dan ook in een gevalideerde korte versie die kan worden gebruikt in ander onderzoek waar kortere testtijden noodzakelijk zijn.

In Hoofdstuk 3 wordt bestudeerd welke van de cognitieve vaardigheden, getest aan het begin van een semesterlang programmeervak, de programmeerprestaties aan het einde van dit semester voorspellen. Hierbij werden programmeervaardigheden op twee manier gemeten: vakgerelateerde programmeervaardigheden, die werden gemeten met de cijfers van de studenten voor het vak, en algemene programmeervaardigheden, die werden gemeten met behulp van de korte versies van de programmeertest zoals beschreven in Hoofdstuk 2. We ontdekten dat logisch redeneren de meest betrouwbare voorspeller was van programmeervaardigheden, aangezien het zowel vakgerelateerde vaardigheden als algemene programmeervaardigheden voorspelde. Algebraïsche vaardigheden en het vermogen om nieuwe woorden te leren voorspelden alleen algemene

programmeervaardigheden. Het leren van grammaticale vaardigheden en patroonherkenning voorspelden geen van beide soorten programmeervaardigheden.

In Hoofdstuk 4 wordt onderzocht of autistische eigenschappen bij bachelorstudenten hun programmeervaardigheden aan het einde van een programmeervak voorspellen. We gebruikten dezelfde deelnemers en dezelfde programmeertests als in Hoofdstuk 3. We ontdekten dat de studenten van dit vak meer autistische eigenschappen hadden dan de gemiddelde bevolking. Autistische eigenschappen hadden echter geen voorspellende waarde voor vakgerelateerde vaardigheden of algemene programmeervaardigheden aan het einde van het vak. De eigenschappen correleerden ook niet met de cognitieve vaardigheden zoals beschreven in Hoofdstuk 3. Het is echter mogelijk dat autistische kenmerken wel verband houden met interesse in programmeren. Dit zou de bovengemiddelde autistische eigenschappen in onze studentenpopulatie verklaren. Toekomstig onderzoek zal deze hypothese verder moeten onderzoeken.

In Hoofdstuk 5 wordt onderzocht of de hersenen fouten in de syntaxis van een programmeertaal (in Java) op een vergelijkbare manier verwerken als grammaticale fouten in hun moedertaal (Nederlands) en een vreemde, natuurlijke taal (Engels). We gebruikten *Event-Related Potentials (ERP's)* als methode om elektrische hersenactiviteit in reactie op stimuli te meten. In alle drie de talen veroorzaakten zinnen met fouten meer positieve hersenactiviteit dan zinnen zonder fouten. Er waren echter wel verschillen tussen de talen met betrekking tot het begin, de duur en de locatie in de hersenen van de effecten. Het effect voor Java begon vroeg, duurde kort en had een frontale en bilaterale verdeling over de schedel. Dit suggereert dat dit soort fouten in de code anders worden verwerkt dan fouten met persoon en getal van werkwoordsvervoeging in de twee natuurlijke talen. Op basis van zowel het tijdsverloop als de topografie van het effect voor Java blijft onduidelijk of dit soort fouten op dezelfde manier verwerkt worden als fouten in natuurlijke talen. Het effect in reactie op de fouten in de programmeersyntaxis lijkt wel op het effect dat in het verleden is waargenomen als reactie op orthografische fouten in een natuurlijke taal. Dit suggereert dat programmeurs de haakjesfouten wellicht als onjuiste spelling beschouwen, in plaats van als onjuiste syntaxis.

Samenvatting

In Hoofdstuk 6 worden de bevindingen van de afzonderlijke studies besproken en worden suggesties gegeven voor toekomstig onderzoek en voor het onderwijs. Op basis van de resultaten van de afzonderlijke onderzoeken lijkt logisch redeneren de meest betrouwbare voorspeller van programmeervaardigheden. Toekomstig onderzoek zal de verbanden tussen cognitieve vaardigheden, onderwijs- en testmethoden, en programmeervaardigheden verder moeten verduidelijken. Daarbij is het vooral van belang om te onderzoeken of taalvaardigheid een grotere rol speelt in programmeercursussen waar syntaxis explicieter wordt onderwezen en beoordeeld. Voor het onderwijs kan het nuttig zijn om logisch redeneren expliciet te onderwijzen. Ten slotte moeten zowel docenten als onderzoekers zich ervan bewust zijn dat, afhankelijk van de gekozen onderwijs- en testmethoden, studenten met bepaalde cognitieve talenten in het voordeel kunnen zijn ten opzichte van anderen.

De bevindingen van dit proefschrift leveren belangrijke nieuwe kennis op over de cognitieve aard van programmeren en bevorderen toekomstig onderzoek op dit gebied. Uiteindelijk zullen de bevindingen in dit onderzoeksveld onze kennis van programmeren als vaardigheid verder vormgeven.

