

Hoe je een betere developer wordt door doelbewust te oefenen

DE KRACHT VAN CODEKATA'S

Wat hebben een goede muzikant en developer met elkaar gemeen? Ze oefenen veel. In andere vakgebieden wordt oefenen als de normaalste zaak van de wereld gezien. Binnen software development gebeurt dat minder. Dat is jammer, want doelbewust oefenen zorgt voor een enorme groei van je vaardigheden. Maar hoe kunnen wij als developers ons vakmanschap effectief oefenen? Een goede manier daarvoor is de CodeKata.

Bespeel jij een instrument? Ik probeer al jaren gitaar te leren spelen en dat is nogal een uitdaging. Van een gevoel voor ritme ontwikkelen, akkoorden leren spelen tot letterlijk eelt kweken op je vingertoppen... het komt allemaal neer op één ding: oefenen. Wees eens eerlijk. Wanneer is de laatste keer dat jij doelbewust hebt geoefend op jouw vaardigheden als developer? Mijn gok is: een tijd geleden...

{ WAAROM CODEKATA'S JE EEN BETERE DEVELOPER MAKEN }

Als ontwikkelaars oefenen we te weinig op de vaardigheden die we nodig hebben tijdens het werk zoals bijvoorbeeld:

- › Naamgeving van variabelen;
- › Bedenken van abstracties;
- › Problemen oplossen met expressieve code.

Natuurlijk komt dit in je werk aan bod, maar er zijn effectievere manieren om die vaardigheden te verbeteren. Denk er maar eens over na.

Het merendeel van de tijd ben je nieuwe functionaliteit aan het bouwen. Mensen wachten op jouw resultaat en je moet alle zeilen bijzetten om dat op tijd te leveren.



V

Roy Straub is Software Engineer & Craftmanship Lead bij CodeSquad en heeft een passie voor vakmanschap in Software Development. Hij schrijft er regelmatig over op zijn blog: www.codecrafter.nl.

Om onder die omstandigheden te leren, is alsof je een nieuw liedje op de gitaar leert spelen terwijl je op het podium staat.

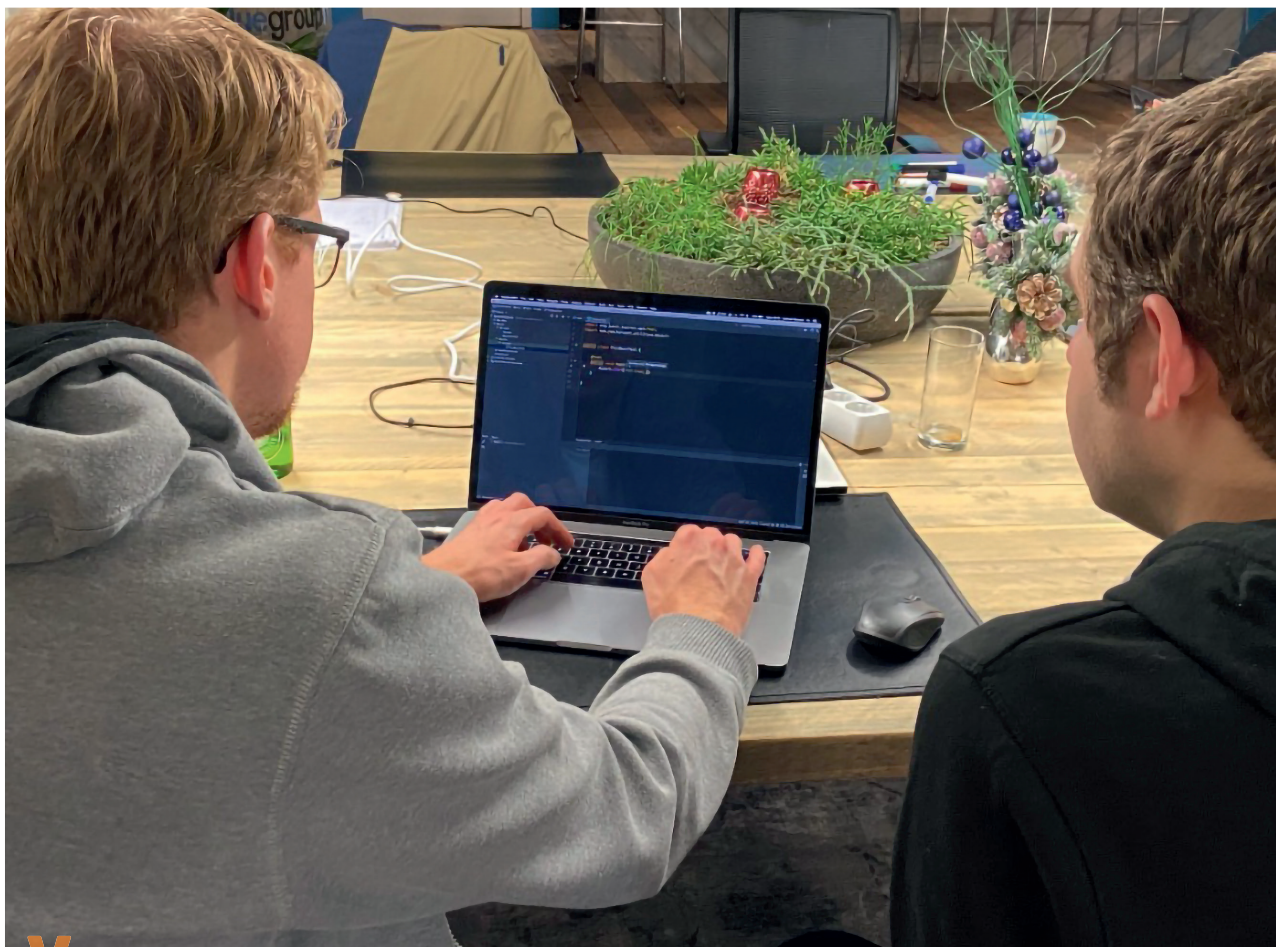
De combinatie van druk en belang bij het resultaat zorgt ervoor dat we minder effectief kunnen leren [1]. We hebben een veilige manier nodig om te kunnen experimenteren en fouten te maken, maar hoe?

{ WAT IS EEN CODEKATA? }

Het idee achter CodeKata's komt uit de vechtsport. Bij een Kata is het de bedoeling dat ingestudeerde bewegingen worden uitgevoerd en de techniek wordt geperfectioneerd [2]. De toepassing van dit concept, binnen software development, is bedacht door Pragmatic Programmer Dave Thomas [3].

CodeKata's zijn (simpele) programmeeropdrachten die je in enkele minuten tot enkele uren kunt oplossen. Net als de Karate Kata kun je deze opdrachten steeds opnieuw doen en er toch van leren. Je perfectioneert je programmeervaardigheden.

Als we terugdenken aan de gitaar zou een Kata bijvoorbeeld een toonladder zijn. Aan wat voor opdrachten kun je dan denken? Bijvoorbeeld het berekenen van de eindscore van een sessie bowlen [4] of het refactoren van een winkel in een fantasy game [5].



V Coding Dojo: samenwerken om je development skills te verbeteren.

De nadruk ligt bij een CodeKata op het verbeteren van je vaardigheden [6]. Dit doe je door doelbewust te oefenen, de uitdaging op te zoeken en feedback te genereren over wat je doet. Stel jezelf dus continu vragen als:

- › Werkt deze oplossing wel of niet?
- › Wordt het hier beter door ja of nee?
- › Kan ik dit ook anders doen?

Reflecteer op wat je doet en leer van je aanpak. Het is belangrijk om te melden dat de uiteindelijke code irrelevant is. De CodeKata is een middel, niet het doel. Wat er wel toe doet, is wat je hebt geleerd tijdens het maken van jouw oplossing. Dus onthoud: Het gaat niet om het resultaat, het gaat om wat je geleerd hebt op je weg ernaartoe.

Als je deze mindset hebt, leg je minder druk op jezelf en dat nodigt uit om te experimenteren en te leren van fouten. Zie het dus vooral als een mooie speeltuin om te oefenen en te ontdekken. Op die manier leer je het meest!

{ WAAROM CODEKATA'S ONMISBAAR ZIJN }

Enkele jaren geleden heb ik mijn eerste CodeKata gedaan. Sindsdien heb ik er aardig wat gemaakt. Ik zie het als een onmisbaar stuk gereedschap om beter te worden in het vak van softwareontwikkeling. Ik benoem de vijf grootste voordelen die ik heb ervaren.

1. Het is jouw feestje

Het mooie van CodeKata's is dat je zelf een leerdoel kunt opstellen en daar doelbewust op kunt oefenen. Jij bepaalt dus zelf wat je leert! Enkele voorbeelden van waar je op kunt oefenen zijn:

- › Test Driven Development toepassen;
- › Effectief Refactoren;
- › Kleine stappen zetten;
- › Clean Code schrijven;
- › SOLID principes toepassen.

De mogelijkheden zijn eindeloos. Als je een CodeKata ziet als een middel, een gereedschap in je gereedschapskist, dan kun je dat voor bijna alles inzetten.

2. Muscle Memory

Bij het leren spelen van gitaar moet je ontzettend veel oefenen op het spelen van verschillende akkoorden en de overgang ertussen. Als je dit vaak genoeg hebt gedaan, hoeft je er niet meer bewust over na te denken en gaat het vanzelf: dat noemen we Muscle Memory.

Ditzelfde effect zal je ervaren op het gebied van programmeren als je veel oefent. Het maken van CodeKata's geeft je "Muscle Memory" bij het programmeren [7]. Dit varieert van het makkelijker uitdrukken van ideeën in code tot het beheersen van de shortcuts van je editor. Oefening en herhaling leiden tot het beheersen van een vaardigheid en dit geldt ook bij software development.

3. Feedback en samenwerking

Het belang van feedback bij leren heb ik benoemd, maar hoe kun je dat het beste genereren? Je kunt CodeKata's uiteraard alleen maken en daar veel van leren, maar er is een betere manier: doe het samen met anderen!

Hoe? Je kunt bijvoorbeeld een CodeKata pair programmeren, mob programmeren of een kata demonstreren aan anderen. Dit samenwerken aan Kata's werkt zo goed dat er zelfs een naam voor is bedacht: een Coding Dojo. Een Coding Dojo is een bijeenkomst specifiek om te oefenen op programmeeropdrachten en je vaardigheden te verbeteren [8]. Bij CodeSquad is dit dé manier waarop we kennis delen.

Het is misschien uitdagend om zo intensief samen te werken met een ander op een stuk code, maar je krijgt veel meer feedback op wat je doet. Daar kan je veel van leren, mits je ervoor open staat. Als je geregeld met verschillende mensen paired, zal je veel andere perspectieven zien!

4. Open Source je portfolio

Is de code waar jij dagelijks aan werkt Open Source? In veel gevallen zal dat helaas niet zo zijn. Dat is jammer, want hoe kan je het werk waar je trots op bent laten zien aan de rest van de wereld?

Als je de oplossing van CodeKata's op een publieke repository zoals Github of Gitlab host, is dat een manier om jouw kunde te delen met de rest van de wereld. Je kan hiermee laten zien in welke onderwerpen jij geïnteresseerd en bedreven bent. Daarnaast kan je je eigen voortgang zien door bijvoorbeeld naar oude code van jezelf te kijken (waarschuwing: dit is af en toe vrij pijnlijk). Je maakt dus je publieke programmeerportfolio!

5. Even wat anders...

Tot slot moeten we even eerlijk zijn. Soms is het werk wat we dag in dag uit doen een beetje saai en ben je toe aan iets anders. Bij een groot project kan je niet zomaar een andere technologie of methodiek gebruiken, alleen om te zien hoe het werkt.

AANGEZIEN KATA'S RELATIEF KORT ZIJN EN EEN DUIDELIJK BEGIN EN EIND HEBBEN MAKEN ZE DINGEN PROBEREN EENVOUDIG

Met een CodeKata kan dit perfect. Wil je weten hoe het is om te programmeren met Clojure? Een keer aan de gang met Test Driven Development? Waarom niet allebei! Aangezien Kata's, in tegenstelling tot projecten, relatief kort zijn en een duidelijk begin en eind hebben maken ze dingen proberen eenvoudig. Sterker nog: je kunt Kata's herhalen waardoor je je meer op het leerpunt kan focussen dan een oplossingsrichting. De oplossing weet je al van de vorige keer.

{ CONCLUSIE }

Een CodeKata is een goed middel om als developer je vaardigheden te verfijnen, zeker als je het gebruikt om doelbewust te oefenen. Als je met enige regelmaat CodeKata's maakt zal je je vaardigheden zien groeien. Het is een perfect middel om te experimenteren en fouten te maken. Hiermee breng je de theorie in de praktijk.

Probeer het eens uit. CodeKata's zijn ook nog eens ontzettend leuk om te doen! Hopelijk heb ik jou ook enthousiast gemaakt voor het doen van een CodeKata. In de bronnenlijst staan enkele sites vernoemd waar je leuke opdrachten kunt vinden. Ik kijk uit naar het maken van de volgende! <

{ REFERENTIES }

- 1 Hunt, A. (2008). Pragmatic Thinking and Learning. Pragmatic Bookshelf.
- 2 Nag, U. (2020, July 16). Everything you need to know to about karate kata <https://www.olympicchannel.com/en/stories/features/detail/karate-kata-martial-arts-history-how-many-forms-meaning-rules-scoring/>
- 3 Thomas, D. (2013, December 31). CodeKata: How It Started. <http://codekata.com/kata/codekata-how-it-started/>
- 4 <https://codingdojo.org/kata/Bowling/>
- 5 <https://codingdojo.org/kata/gilded-rose/>
- 6 Thomas, D. (2014, April 14). CodeKata. <http://codekata.com/kata/codekata-intro/>
- 7 Martin, M. (2020, October 14). Code Kata Consequences. <https://cleancoders.com/blog/2020-10-14-code-kata-consequences>
- 8 <https://codingdojo.org/WhatIsCodingDojo/>
- 9 <https://codingdojo.org/kata/>
- 10 <http://codekata.com/>
- 11 <https://kata-log.rocks/>