



Boom

Guido Dulos

PROGRAMMEREN IN JAVA MET SCRUM

Programmeren in Java met Scrum

Guido Dulos

Boom

**inclusief
website!**

Met behulp van onderstaande unieke activeringscode krijg je toegang tot de website www.javametscrum.nl voor extra materiaal. Deze code is persoonsgebonden en gekoppeld aan de 1^{ste} druk. Na activering van de code is de website twee jaar toegankelijk. De code kan tot zes maanden na het verschijnen van een volgende druk geactiveerd worden. De code is eenmalig te gebruiken.

Opmaak binnenwerk: Holland Graphics, Amsterdam
Basisontwerp omslag: Dog & Pony, Amsterdam
Omslagontwerp: Coco Bookmedia, Amersfoort
Beeld omslag: Paul Bradbury/iStock

© Guido Dulos & Boom uitgevers Amsterdam, 2021

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden veelevoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor het overnemen van (een) gedeelte(n) uit deze uitgave in bijvoorbeeld een (digitale) leeromgeving of een reader in het onderwijs (op grond van artikel 16, Auteurswet 1912) kan men zich wenden tot Stichting Uitgeversorganisatie voor Onderwijslicenties, Postbus 3060, 2130 KB Hoofddorp, www.stichting-uvo.nl.

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 9789024429745

ISBN e-book 9789024429752

NUR 123

www.javametscrum.nl

www.boomhogeronderwijs.nl

Inhoudsopgave

Voorwoord	11
Dankwoord	13
Tips om dit boek te gebruiken	15
DEEL 1: JAVA LEREN MET SCRUM	17
1 De voorbereiding	19
1.1 Productvisie	21
1.2 Product backlog	21
1.3 Inrichten van de omgeving	21
1.4 Way of Working (WoW)	21
2 Sprint 1	25
2.1 Sprintplanningsmeeting 1	25
2.2 Sprintplanningsmeeting 2	29
2.3 Realisatie	29
2.3.1 <i>Daily scrum</i>	29
2.3.2 <i>De realisatie van user story US01</i>	30
2.3.3 <i>De realisatie van user story US02</i>	32
2.3.4 <i>Daily scrum</i>	34
2.3.5 <i>Mutation testing</i>	34
2.3.6 <i>De realisatie van user story US03</i>	35
2.4 Sprintreview	38
2.5 Retrospective	39
3 Sprint 2	43
3.1 Inleiding	43
3.2 Sprintplanningsmeeting 1	43
3.2.1 <i>Planning poker</i>	44
3.3 Sprintplanningsmeeting 2	44
3.4 Realisatie	44
3.4.1 <i>Het opstellen van testcases voor de functionaliteit van sprint 1</i>	44
3.4.2 <i>Statement coverage</i>	47
3.4.3 <i>De realisatie van user story US04</i>	48
3.4.4 <i>De realisatie van user story US05</i>	50
3.4.5 <i>Testen van interacties tussen gebruiker en systeem</i>	53

3.4.6	<i>De realisatie van user story US06</i>	53
3.5	Sprintreview	56
3.6	Retrospective	56
4	Sprint 3	59
4.1	Sprintplanningsmeeting 1	59
4.2	Sprintplanningsmeeting 2	60
4.3	Realisatie	60
4.3.1	<i>De realisatie van user story US07</i>	60
4.3.2	<i>Daily scrum</i>	62
4.3.3	<i>De realisatie van user story US08</i>	63
4.3.4	<i>De realisatie van user story US09</i>	63
4.3.5	<i>Spike</i>	64
4.3.6	<i>Daily scrum</i>	65
4.3.7	<i>De realisatie van user story US10</i>	65
4.3.8	<i>Informatie over een class en JUnit testen</i>	70
4.3.9	<i>Gebruikerstesten</i>	71
4.4	Sprintreview	71
4.5	Retrospective	72
5	Sprint 4	75
5.1	Sprintplanningsmeeting 1	75
5.2	Sprintplanningsmeeting 2	76
5.3	De realisatie	78
5.3.1	<i>De realisatie van user story US11</i>	78
5.3.2	<i>De realisatie van user story US12</i>	81
5.3.3	<i>De realisatie van user story US13</i>	82
5.3.4	<i>De realisatie van user story US14</i>	88
5.3.5	<i>De realisatie van user story US15</i>	90
5.4	Sprintreview	92
5.5	Retrospective	94
6	Sprint 5	97
6.1	Inleiding	97
6.2	Sprintplanningsmeeting 1	97
6.3	Sprintplanningsmeeting 2	98
6.4	De realisatie	99
6.4.1	<i>Spike02</i>	99
6.4.2	<i>Realisatie van user story US16a</i>	100
6.4.3	<i>Realisatie van user story US16b</i>	104
6.4.4	<i>Realisatie van user story US16d</i>	106
6.4.5	<i>Realisatie van user story US16e</i>	108
6.4.6	<i>Realisatie van user story US16f</i>	109
6.4.7	<i>Realisatie van user story US16g</i>	110
6.4.8	<i>Realisatie van user story US16h</i>	110

6.4.9	<i>Realisatie van user story US16i</i>	113
6.4.10	<i>Testen op basis van statement coverage</i>	114
6.4.11	<i>Realisatie van user story US16c</i>	114
6.4.12	<i>Realisatie van user story US16j</i>	116
6.5	Sprintreview	116
6.6	Sprintretrospective	116
7	Sprint 6	119
7.1	Sprintplanningsmeeting 1	119
7.2	Sprintplanningsmeeting 2	120
7.3	De realisatie	120
7.3.1	<i>De realisatie van user story US17</i>	120
7.3.2	<i>De realisatie van user story US18</i>	122
7.3.3	<i>De realisatie van user story US19</i>	124
7.3.4	<i>De realisatie van user story US20</i>	130
7.4	Sprintreview	132
7.5	Sprintretrospective	132
8	Sprint 7	135
8.1	Sprintplanningsmeeting 1	135
8.2	Sprintplanningsmeeting 2	136
8.3	De realisatie	136
8.3.1	<i>De realisatie van user story US21</i>	136
8.3.2	<i>De realisatie van user story US22</i>	138
8.3.3	<i>De realisatie van user story US23</i>	140
8.4	Sprintreview	144
8.5	Sprintretrospective	144
9	Sprint 8	147
9.1	Inleiding	147
9.2	Sprintplanningsmeeting 1	147
9.3	Sprintplanningsmeeting 2	147
9.4	Spike	149
9.5	Sprintreview	149
9.6	Sprintretrospective	149
	DEEL 2: THEORIEGEDEELTE SCRUM EN JAVA	151
10	Het framework scrum	153
10.1	Wat is scrum?	153
10.2	Waarom scrum?	153
10.3	Wat zijn de kenmerken van scrum?	154
10.4	Wat is de relatie tussen scrum en andere agile methodes?	154
10.5	Welke rollen zijn er binnen scrum?	155
10.6	Welke events of gebeurtenissen kent scrum?	156

10.7	Wat is de voorbereidingsperiode?	156
10.8	Wat zijn de kenmerken van een sprint?	157
10.9	Wat gebeurt er tijdens sprintplanningsmeeting 1?	157
10.10	Wat gebeurt er tijdens sprintplanningsmeeting 2?	157
10.11	Wat is de ontwikkelperiode?	157
10.12	Wat is de daily scrum?	158
10.13	Wat gebeurt er tijdens de sprintreview?	158
10.14	Wat is de sprintretrospective?	158
10.15	Hoe verloopt een refinementsessie?	158
10.16	Hoe verloopt planning poker?	159
10.17	Welke producten en hulpmiddelen (artefacten) spelen bij scrum een rol?	160
10.18	Wat is een incremental?	160
10.19	Wat is de product backlog?	160
10.20	Hoe werkt het team met items (thema's, epics en user stories)?	161
10.21	Hoe ziet de sprint backlog eruit?	162
10.22	Wat is de functie van de Definition of Ready (DoR)?	162
10.23	Wat is het belang van de eis dat een user story INVEST moet zijn?	162
10.24	Wat is een spike?	163
10.25	Wat is refactoring?	163
10.26	Wat is de Definition of Done (DoD)?	164
10.27	Welke visualisaties in de werkruimte kan een team aanbren- gen?	164
10.28	Welke rol speelt automatisering binnen scrum?	165
10.29	Welke kenmerken en waarden heeft een succesvol scrum- team?	165
10.30	Wat is werken volgens een sustainable pace?	165
10.31	Kom ik als ontwikkelaar zeker in een scrumteam?	165
10.32	Wat is personal scrum?	166
11	De programmeertaal Java	169
11.1	Wat is een computerprogramma?	169
11.2	Hoe ziet Javacode eruit en hoe kun je deze lezen?	170
11.3	Hoe schrijf je een computerprogramma?	170
11.4	Wat is er nodig om een computerprogramma te laten werken?	171
11.5	Welke programma's en tools zijn nodig om te programmeren?	172
11.6	Hoe verloopt het programmeerproces?	172
11.7	Is Java 100% hoofdlettergevoelig?	173
11.8	Uit welke elementen bestaat Javacode?	173
11.9	Wat is een Javafile?	179
11.10	Hoe is een class in Java gestructureerd?	179
11.11	Welke regels en conventies gelden er voor de namen van identifiers?	180

11.12	Welke conventies zijn er ten aanzien van de opmaak van de code?	180
11.13	Op welke vijf manieren doorloopt Java een programma?	181
11.14	Hoe werkt System.out.println(), System.out.print(), System.out.printf()?	182
11.15	Wat is een literal?	184
11.16	Wat is een variabele?	185
11.17	Wat is een method?	186
11.18	Wat is de betekenis van de method main()?	188
11.19	Wat zijn primitieve variabelen?	189
11.20	Wat is het bereik/de scope en levensduur van een variabele?	190
11.21	Welke rekenmogelijkheden en rekenregels zijn er in Java?	191
11.22	Hoe werk je in Java met decimale getallen en hoe werkt casting?	194
11.23	Hoe werk je in Java met if-statements en switch-case?	194
11.24	Hoe werkt Java met vergelijkingen?	198
11.25	Wat zijn referentievariabelen?	199
11.26	Hoe werk je met objecten in Java?	200
11.27	Hoe ziet een <i>objectclass</i> eruit?	202
11.28	Wat is de betekenis van = bij primitieve variabelen en referentievariabelen?	206
11.29	Hoe achterhaal je de eigenschappen en mogelijkheden van een object?	206
11.30	Hoe werk je met objecten van het type String?	209
11.31	Wat zijn nuttige methods voor objecten van het type String?	210
11.32	Wat zijn nuttige methods voor objecten van het type Scanner?	211
11.33	Wat zijn nuttige methods voor objecten van het type LocalDate?	213
11.34	Hoe werk je met iteraties?	214
11.35	Hoe werk je met arrays?	216
11.36	Hoe werk je met ArrayLists?	217
11.37	Hoe kun je gegevens opslaan op schijf en weer ophalen?	219
11.38	Wat zijn exception errors en hoe vang je ze af met try-catch?	221
11.39	Wat betekenen de begrippen overerving, polymorfisme en abstracte class?	225
11.40	Hoe bereik je polymorfisme met Interfacing?	232
12	Werken met de grafische gebruikersinterface (AWT en Swing)	237
12.1	Wat moet ik weten voordat ik een gebruikersinterface ga ontwerpen?	237
12.2	Wat is AWT, Swing en JavaFX?	240
12.3	Wat zijn GUI-componenten?	241
12.4	Waar vind ik documentatie over alle GUI-componenten?	241
12.5	Wat is een lay-outmanager?	242
12.6	Wat zijn states?	243

12.7	Hoe werkt een GUI met triggers, listeners en events?	245
12.8	Welke soorten listeners zijn er?	246
12.9	Hoe implementeer ik listeners en de afhandeling van events?	246
12.10	Hoe codeer je listeners en eventhandlers?	247
12.11	Wat is een pattern of programmeerprincipe?	249
12.12	Wat is het basisidee van het MVC-principe?	250
12.13	Waarop moet ik letten bij MVC?	251
12.14	Hoe maak ik een view?	252
12.15	Hoe maak ik een model?	252
12.16	Hoe maak ik een controller?	253
12.17	Hoe kan ik een GUI-applicatie testen?	253
12.18	Hoe ziet de main method eruit in een GUI-MVC-applicatie?	253
12.19	Hoe ziet een GUI-programma eruit dat met behulp van een MVC-architectuur is gemaakt?	254
DEEL 3: INSTALLEREN EN TESTEN		263
13	Installeren van en werken met de JDK en IntelliJ	265
13.1	Installeren van de ontwikkelomgeving IntelliJ (inclusief Git, Gradle en JUnit)	265
13.2	IntelliJ verbinden met de JDK	265
13.3	Way of Working met IntelliJ, Git, Gradle en JUnit	266
13.4	Tips voor het gebruik van IntelliJ	271
14	Testen binnen IntelliJ	275
14.1	Wat is testen?	275
14.2	Wat zijn testdoelen en wat is de relatie met testcases?	275
14.3	Wat zijn unittesten?	276
14.4	Wat zijn unitintegrietesten?	277
14.5	Hoe ziet de testomgeving eruit?	277
14.6	Wat is JUnit?	278
14.7	Hoe werk je met JUnit in IntelliJ?	278
14.8	Hoe maak ik testcases in main?	282
Index		285

Voorwoord

Agile ontwikkelmethodieken zoals scrum zijn niet meer weg te denken bij het maken van nieuwe software. Belangrijk kenmerk hiervan is dat je in kleine stappen nieuwe programmatuur maakt. In dit boek leer je de programmataal Java kennen en toepassen met behulp van deze methodiek. Dat betekent dat je niet alleen leert coderen, maar er ook veel aandacht is voor de eisen en wensen van de klant. Hoe die boven tafel te krijgen, het uitvoeren van een businessanalyse, ontwerpen, documenteren, testen, samenwerken, communiceren en... fouten maken en daarvan leren.

De beste manier om Java te leren is door het 'te doen'. De scrumomgeving is daarvoor een ideale omgeving. Daarom heeft dit boek een praktijksituatie als uitgangspunt: je werkt als teamlid van een scrumteam aan een Javaproject. Je leert daardoor in een hoog tempo Java.

Het mooiste is als je in teamverband kunt werken aan het bedenken en realiseren van het programma dat je met behulp van dit boek stap voor stap gaat ontwikkelen. Maar ook als je het boek zelfstandig doorwerkt, kun je daarbij de scrumprincipes hanteren. Je vormt dan een scrumteam-in-je-eentje. Dat noemen we personal scrum of scrum for one.

De opbouw van het boek is als volgt:

In deel 1 maak je gedurende een voorbereidingsfase en acht sprints deel uit van een scrumteam. Je werkt met dit team aan een desktopapplicatie voor een klant. Jij bent een van de ontwikkelaars. Een speciaal teamlid helpt het team met het leren begrijpen en toepassen van Java. Met het complexer worden van de applicatie groeien jouw kennis van de programmeertaal Java en je vaardigheden als Java-ontwikkelaar in een scrumteam.

Deel 2 bevat de theorie van scrum en Java. De hoofdstukken daarvan vormen samen het naslagwerk dat je gebruikt om het project uit het boek te realiseren. Dat naslagwerk zal straks een belangrijk hulpmiddel zijn voor de projecten die je gaat uitvoeren als je dit boek uit hebt!

Deel 3 bevat de installatiehandleiding voor de software die je nodig hebt en een introductie van deze software. Je werkt in het project met Javaversie 8 of hoger en de gratis ontwikkeltool IntelliJ Community Edition. Het programma trello gebruik je om te communiceren over de user stories – de specificaties – en de voortgang. Deze programma's zijn beschikbaar voor Windows, MacOS en Linux.

Een belangrijk kenmerk van scrum is dat het team continu leert. En continu leren betekent: veel vragen stellen en zorgen dat je antwoorden op die vragen vindt. Om die reden is het naslagwerkgedeelte helemaal opgebouwd aan de hand van vragen. De volgorde van die vragen is zo veel mogelijk gelijk gehouden aan de groeiende kennis die je nodig hebt per sprint.

Tot slot van deze inleiding een uitleg van de keuzes die ik als auteur voor dit boek heb gemaakt.

- Een programmeur is niet alleen maar ‘codeur’. Vandaar dat dit boek je opleidt tot een T-shaped ontwikkelaar, die veel afweet van de taal Java, maar ook de andere benodigde vaardigheden goed beheerst.
- Er is slechts één case. Dat doet recht aan de wijze waarop je straks in je werk in een scrumteam werkt. Je zult dan gedurende vele sprints aan één applicatie werken. Het onderhouden van je geautomatiseerde testen is dan heel belangrijk en ook het belang van refactoren wordt duidelijk. Door het werken met één case word je ook steeds meer ‘domeindeskundige’ en stijgt jouw waarde voor het team. Dat zijn allemaal zaken die je gedurende dit project gaat ervaren.
- Java is een complexe taal, met zeer veel mogelijkheden. Dit boek is een inleiding in de taal, waarbij de belangrijkste aspecten aan de orde komen. Na bestudering van dit boek en het uitvoeren van het project ben je in staat om eigen programma’s te maken en je weg te vinden om met de documentatie op internet je kennis verder uit te breiden.
- De code bevat geen commentaar. De uitleg krijg je in de tekst.
- In het bedrijfsleven zul je voornamelijk je code in het Engels schrijven. Omdat dit een boek is voor de Nederlandstalige markt, is het besluit genomen om in de code Nederlandse termen te gebruiken.
- Alle broncode is te vinden op de website die bij dit boek hoort. Je vindt daar ook voorbeelden van de code in het Engels, inclusief comments.
- Op de website vind je extra oefeningen, verdieping en multiplechoicevragen.



Veel plezier bij het programmeren met Java in scrum!

Guido Dulos (maart 2021)

Dankwoord

Het idee voor dit boek is ontstaan tijdens het verzorgen van trainingen over Java, scrum en testen. Dank aan de cursisten van deze trainingen, die mij geholpen hebben het idee om Java te leren in een scrumcontext verder uit te diepen. Dank ook aan het meeleesteam van de uitgever en Anja van Harskamp voor hun vele waardevolle tips. Speciale dank aan Loes Heiligers, van uitgeverij Boom, die gedurende het hele schrijftraject steeds snel en duidelijk mijn vele vragen beantwoordde.

Tips om dit boek te gebruiken



Deel 1 beschrijft een doorlopende case, waarbij een team binnen een scrumproject ontwikkeling in Java leert. Deel 2 bevat de belangrijkste theorie van Java, met veel voorbeelden. Deel 3 geeft stappenplannen voor het installeren en werken met de softwaretools. Het is belangrijk dat je niet alleen leert coderen, maar tevens leert denken als een agile ontwikkelaar. Dat is ook de reden waarom het leren van Java in dit boek is ingebed in de scrumwerkwijze.

Scrum is ingedeeld in vaste periodes, sprints genoemd. In die sprints werkt het scrumteam aan een programma. Het doel van een sprint is dat het programma waaraan het team werkt, na de sprint voor de klant waardevoller is geworden. Voor elke sprint kiest het team een aantal wensen van de klant en gaat dan aan de slag met het programma om ervoor te zorgen dat het programma aan het eind van de sprint aan die wensen voldoet. Het mooie is dat de klant al na de eerste sprint een werkbaar programma heeft. In elke opvolgende sprint voegt het team nieuwe functionaliteit toe zodat het programma steeds meer voldoet aan de verwachtingen van de klant. Elke wens van de klant is beschreven in een speciale vorm: de user story.

In dit boek realiseer je in 8 sprints 23 van deze user stories. Daarbij leer je bij elke user story iets nieuws over Java.

Hier volgen wat tips om op zo'n manier met dit boek te werken dat je snel in je rol van ontwikkelaar groeit:

- Dit boek is vooral een werkboek. Werk in deel 1 sprint voor sprint, user story voor user story aan het ontwerpen en coderen van het beschreven Java-programma.
- Bedenk bij elke nieuwe user story eerst hoe je deze zou coderen en testen met de kennis die je van Java hebt. Ga ook na welke kennis je nog niet hebt.
- Elke sprint en elke user story begint met een beschrijving wat je gaat maken en leren. Bestudeer de aangegeven paragrafen van deel 2 grondig, ook door zelf de voorbeelden te coderen.
- Ga met je nieuwe kennis aan de slag om de user stories te realiseren. Dankzij de versiebeheertool Git kun je verschillende versies maken.
- Bestudeer de code in het boek en neem deze over in je programmeertool. Voorzie die code eventueel van commentaar.

-  Leg een logboek aan, waarin je ‘snippets’ (stukjes code) opneemt waarvan je denkt dat je die later kunt hergebruiken. Op de website vind je een voorbeeld van zo’n logboek.
- Maak op basis van je testdoelen geautomatiseerde testen (in JUnit, en in een opstartclass).
-  Bedenk zelf een eigen project met user stories en realiseer die met de kennis die je hebt, de theorie uit deel 2 en de verdieping op de website.


Java leren met scrum

DEEL 1

In dit deel ben je een aantal weken ontwikkelaar in een scrumteam. Je ontwikkelt een applicatie die steeds waardevoller wordt voor de opdrachtgever. Terwijl je stap voor stap de wensen van de klant realiseert, maak je kennis met de scrumcultuur en leer je programmeren in Java.

De volgorde waarin je de concepten van Java leert, hebben te maken met de applicatie en met 'wat de klant wil'. Dat houdt bijvoorbeeld in dat je al vanaf sprint 1 met objecten leert werken en vrij snel kennismaakt met arrays en ArraysLists. Termen die je nu misschien nog niet veel zeggen, maar die in veel leerboeken pas na zo'n honderd bladzijden aan bod komen. Je leert theorie veel sneller als je die nodig hebt om je doel te bereiken. Deze manier van leren komt overeen met de filosofie van scrum: je leert iets door het toe te passen.

Een wat meer systematische beschrijving van scrum en van Java vind je in deel 2 van dit boek. Bij elke sprint en elke user story lees je welke paragrafen van deel 2 je helpen de benodigde Javaconcepten te begrijpen. Door die stof te bestuderen en tegelijk het project uit te voeren krijg je een gedegen kennis van de taal.

Je wordt van harte uitgenodigd om tijdens de sprint op internet te zoeken naar de betekenis van Javaconcepten en hoe je die kunt toepassen. Verder leer je Java door na te denken, te praten met je teamgenoten, dingen uit te proberen en dit boek en andere bronnen te raadplegen. Op de website die bij dit boek hoort, vind je nog meer materiaal dat je helpt Java steeds beter te begrijpen. 

Om je de sfeer van een scrumomgeving te laten proeven, zijn hoofdstuk 1 – de voorbereiding – en hoofdstuk 2 – sprint 1 – in storytelvorm geschreven. Deze hoofdstukken bevatten dialogen van de teamleden. Vanaf hoofdstuk 3 zijn deze dialogen weggelaten, waarbij er wel aandacht blijft voor de uitdagingen van het scrumteam.



De voorbereiding

1

Lees ook	Onderwerp
Hoofdstuk 10	Het framework scrum
Hoofdstuk 13	Installeren van en werken met de JDK en IntelliJ

Vanaf nu ben je ontwikkelaar in een scrumteam. Net als jij hebben de meeste andere scrumleden weinig of geen kennis van scrum en Java. Maar dat zal snel veranderen!

Carla is scrummaster en heeft al ervaring met werken volgens scrum. Marc heeft ervaring met Java en zal tijdens de sprints het team op weg helpen om stap voor stap de taal te leren beheersen.

Het scrumproject start met de voorbereiding, ook wel sprint 0 genoemd. Je maakt kennis met je teamleden en krijgt informatie over scrum en de applicatie die je gaat maken. Je installeert de software die nodig is om je ontwikkelwerk te doen en maakt afspraken met elkaar over de wijze waarop jullie gaan werken.

De voorbereiding begint met een bijeenkomst voor het hele team. Product-owner Koen, scrummaster Carla en de ontwikkelaars: Marc, Peter, Mohammad, Youssef, Shayla en jij.

Marc

‘Welkom in het scrumteam! Met dit scrumteam ontwikkelen we de komende weken een applicatie voor de klant. Koen zal hier straks meer over vertellen. Eerst geeft Carla wat uitleg over scrum.’

Carla

‘Welkom allemaal! Aan mij de eer om jullie uit te leggen wat scrum is. Daar ben ik vrij snel mee klaar. Ken Schwaber, een van de twee bedenkers van scrum, zegt over dit framework: “Eenvoudig te begrijpen, moeilijk te beheersen.” Om jullie uit te leggen wat scrum is, heb ik maar een uurtje nodig. De maanden daarna hebben we nodig om steeds beter in scrum te worden.’

In het uur dat volgt legt Carla aan de hand van een aantal vragen het scrum-framework uit (zie hoofdstuk 10).

Na de koffiebreek maakt Carla haar verhaal af:

‘Ik gaf aan dat scrum een aantal strakke regels kent én daarnaast een enorme vrijheid biedt. De komende weken zullen we kijken hoe scrum voor ons werkt en aanpassingen maken, waar nodig én toegestaan.

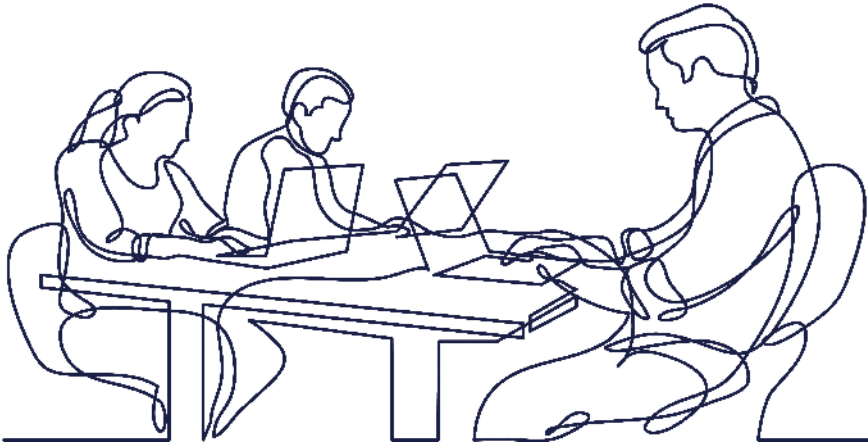
Ik heb op dit moment al een voorstel voor een aanvulling. Omdat continu leren zo ontzettend belangrijk is, wil ik graag dat we naast de product backlog en de sprint backlog een derde bord gebruiken: de leerbacklog. Daarin maken we zichtbaar wat we willen leren en wat we hebben geleerd. De scrumpresentatie heb ik gegeven aan de hand van vragen en antwoorden. Voor het leren van Java kunnen we dezelfde vraag- en antwoordstructuur hanteren. We kunnen van elke vraag een kaart maken die we op de leerbacklog plaatsen.’

Shayla

‘Ik vind dit een goed idee. De afgevinkte vragen laten onze voortschrijdende kennis zien en de vragen die nog niet zijn afgevinkt onze kennisbehoefte.’

Mohammad

‘Afvinken kan, maar we kunnen ook twee kolommen op het bord maken: “leren” en “geleerd”. Dat geeft nog duidelijker aan hoe ver we zijn en het lijkt meer op de structuur van de andere twee borden.’



Het team gaat akkoord. Het gebruik van de leerbacklog is hiermee onderdeel van de Way of Working (WoW) van het team.

Ook een tweede voorstel van Carla vindt unaniem bijval en behoort vanaf dat moment tot de werkwijze van het team: Marc zal bij elke sprint, vooral tijdens de sprintplanningsmeetings, uitleg geven over Javaconcepten die nodig zijn om de items in de sprint te kunnen realiseren.

Carla

‘Vanmiddag zal Koen uitleggen welke applicatie we gaan maken. Daarna gaat Marc ons helpen bij het installeren van de benodigde software en het inrichten van het project. En morgen beginnen we aan sprint 1.’

1.1 Productvisie

Koen

‘We gaan voor een school in Amersfoort een hulpmiddel ontwikkelen waarmee de scholieren zich beter kunnen voorbereiden op repetities en examens. Het hart van dat hulpmiddel wordt een flashcardprogramma. Een flashcard of leerkaart is een kaartje waar op de voorkant een vraag staat en op de achterkant het antwoord. Je kunt met een set flashcards bijvoorbeeld Engelse woordjes leren. De flashcards zijn ook voor andere vakken geschikt. Flashcards kun je zelfs gebruiken om samenvattingen te maken, of om werkstukken te generen.

Dit project is erg geschikt om met scrum aan te pakken. We beginnen klein en kijken wel waar we uitkomen. Elke sprint leveren we een werkend product op. Het mooie is dat de klant het geweldig leuk vindt dat ze met een team samenwerkt dat alles nog moet leren. Vandaar dat wij deze opdracht hebben gekregen.’

1.2 Product backlog

Carla

‘De klant heeft een aantal thema’s in gedachten. De komende sprints werken wij aan een van de thema’s: het ontwikkelen van een flashcardprogramma. Koen heeft een trello-bord ingericht waarop al wat wensen van de klant staan. Die bekijken we morgen als we met sprint 1 starten. We maken van de items die we het eerst willen oppakken behapbare user stories.’

1.3 Inrichten van de omgeving

Marc

‘Eerst ga ik jullie helpen bij het installeren van de benodigde software.’

Na installatie van de software kan het team uit de startblokken. Marc geeft het team een inleiding in het werken met IntelliJ, Gradle, JUnit, Git en het trello-bord. Het team wil graag de sprint backlog inrichten, maar dat is toch pas iets voor de volgende dag.

In hoofdstuk 13 lees je welke software je nodig hebt en hoe je deze kunt installeren.

Je krijgt ook informatie over hoe je je project inricht en tips hoe je met de software kunt werken.



1.4 Way of Working (WoW)

Als laatste stap van de voorbereiding buigt het team zich over de Way of Working. Ze bespreken onder andere de duur van de sprints. Koen stelt voor om voor de sprintduur een week te kiezen.

Agile ontwikkelmethoden zoals scrum zijn niet meer weg te denken bij het maken van nieuwe software. In dit boek leer je de programmeertaal Java kennen en toepassen met behulp van deze methodiek. De beste manier om Java te leren is door het te doen en de scrumomgeving is daarvoor ideaal. Eén van de belangrijke onderdelen van scrum is immers het 'continu leren en continu reflecteren'.

Je leert programmeren in de context van de echte wereld. Praktische toepassing en theoretisch inzicht gaan hand in hand met behulp van een casus die door het hele boek heen loopt.

Naast de focus op het leren coderen, is er eveneens veel aandacht voor alle andere zaken waar je binnen een project mee te maken krijgt. Denk hierbij aan het helder krijgen van de eisen en wensen van de klant en het uitvoeren van business analyses. Alle aspecten van het maken van een product komen aan de orde: ontwerpen, documenteren, testen, samenwerken, communiceren en ook het maken van fouten en daar weer van leren.

Het boek volgt een project van een scrumteam gedurende een aantal sprints. Binnen dit project maakt het team zich tevens de taal Java eigen, waarbij regelmatig wordt verwezen naar de theorie in de twee laatste delen. Als lezer maak je onderdeel uit van dit scrumteam en denk je mee over het vertalen van de wensen van de klant in werkende code. *Programmeren in Java met Scrum* is hierdoor een goede voorbereiding op de arbeidsmarkt.

Bij dit boek hoort de website www.javametscrum.nl waar je kunt oefenen met een vaardigheidstoets, meerkeuzevragen en open vragen aan de hand van user stories. Ook is de source code die in het boek voorkomt hier te vinden.

Dit boek is geschikt voor programmeercursussen binnen het hoger onderwijs en iedereen die binnen een scrumomgeving wil leren programmeren in Java.

Guido Dulos is sinds 1987 werkzaam in de ICT en verzorgt als docent trainingen op het gebied van kwaliteitszorg, testen, communicatie, Java en C#.

