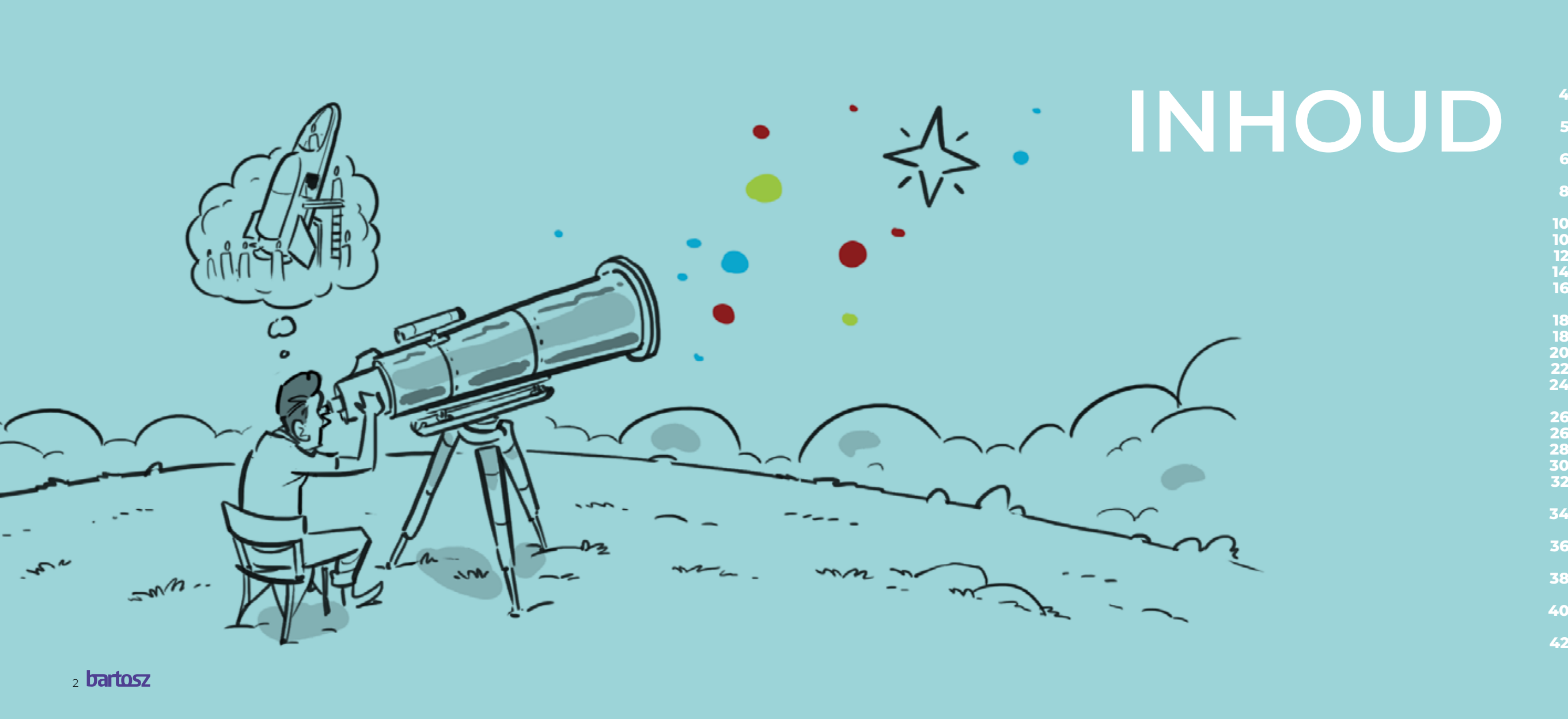


IMPROVE YOUR BEST TEAM

EEN REISGIDS VOOR QUALITY INFECTED TEAMS





INHOUD

4	OVER DIT BOEKJE
5	ONZE VISIE OP KWALITEIT
6	DE DRIE PIJLERS
8	QIT IN DE PRAKTIJK
10	1. WHOLE TEAM APPROACH
10	1.1 ANNES PROBLEEM
12	1.2 LAYLA'S ANALYSE
14	1.3 TIPS
16	1.4 THEORETISCH KADER
18	2. FAST FEEDBACK
18	2.1 ANNES PROBLEEM
20	2.2 LAYLA'S ANALYSE
22	2.3 TIPS
24	2.4 THEORETISCH KADER
26	3. EXPLORATION
26	3.1 ANNES PROBLEEM
28	3.2 LAYLA'S ANALYSE
30	3.3 TIPS
32	3.4 THEORETISCH KADER
34	EEN SUCCESVOLLE DAG
36	HOE NU VERDER ALS TEAM
38	HOE NU VERDER ALS PROFESSIONAL
40	MIJN REISNOTITIES
42	TOT SLOT

OVER DIT BOEKJE

Improve your best! Dat is waar we bij Bartosz in geloven. We dagen onszelf, en ook jou, graag uit om continu op zoek te gaan naar verbetering. Bij alles wat we doen stellen we de vraag: kan het iets sneller, iets slimmer, iets beter?

We hebben onszelf als doel gesteld om het meest innovatieve testbedrijf van Nederland te zijn en te blijven. Daarbij speelt ons innovatieplatform Bartosz Labsz een grote rol. Tijdens maandelijkse bijeenkomsten experimenteren we met nieuwe ideeën om zo sneller op de veranderende markt te kunnen inspelen. Elke Bartoszian kan zijn of haar bijdrage leveren aan die experimenten. Dit boekje is het resultaat van zo'n experiment.

Wellicht vraag je je af waarom een testbedrijf een boekje schrijft over het verbeteren van teams. Een tester werkt niet alleen, maar is onderdeel van een team dat gezamenlijk een product bouwt. Waar het testwerk vroeger vaak tot het einde bewaard werd, gaat het nu gelijk op met het ontwikkelen. Het hele team, waaronder testers en ontwikkelaars, werkt samen om een kwalitatief goed product neer te zetten. Een product dat voor de gebruikers echt waardevol is. Wij willen er voor zorgen dat die teams 'Quality Infected Teams' worden: teams die in een voorspelbaar tempo software van de juiste kwaliteit leveren. Kwaliteit is immers niet de verantwoordelijkheid van een enkel teamlid, maar van het hele team.



ONZE VISIE OP KWALITEIT

Testen is bij ons nooit een doel op zich. We verzamelen feedback. Het team gebruikt continu feedback om de kwaliteit van het product te bewaken en te verbeteren. Een product maken dat echt waardevol is voor de gebruiker, daar draait het om! Testen staat dus nooit op zichzelf. Vandaar ook de naam van onze visie: Quality Infected Teams (QIT).

Als team werk je samen om het doel te bereiken: een kwalitatief hoogwaardig product. Onderweg naar die bestemming kom je uitdagingen tegen. Uitdagingen die heel divers kunnen zijn: van elkaar niet goed begrijpen en technische problemen tot onverwachtse gebeurtenissen. Binnen QIT hebben we drie aandachtsgebieden geïdentificeerd waar deze uitdagingen vaak naar te herleiden zijn: de Whole Team Approach, Fast Feedback en Exploration. Deze noemen we de drie pijlers van QIT.

DE DRIE PIJLERS

Deze drie pijlers categoriseren uitdagingen die ontwikkelteams tegenkomen. In de praktijk vallen gebeurtenissen met impact op kwaliteit vaak niet binnen één van de pijlers, maar is er overlap. Een Quality Infected Team verbetert zich actief binnen alle drie de pijlers.

1. WHOLE TEAM APPROACH

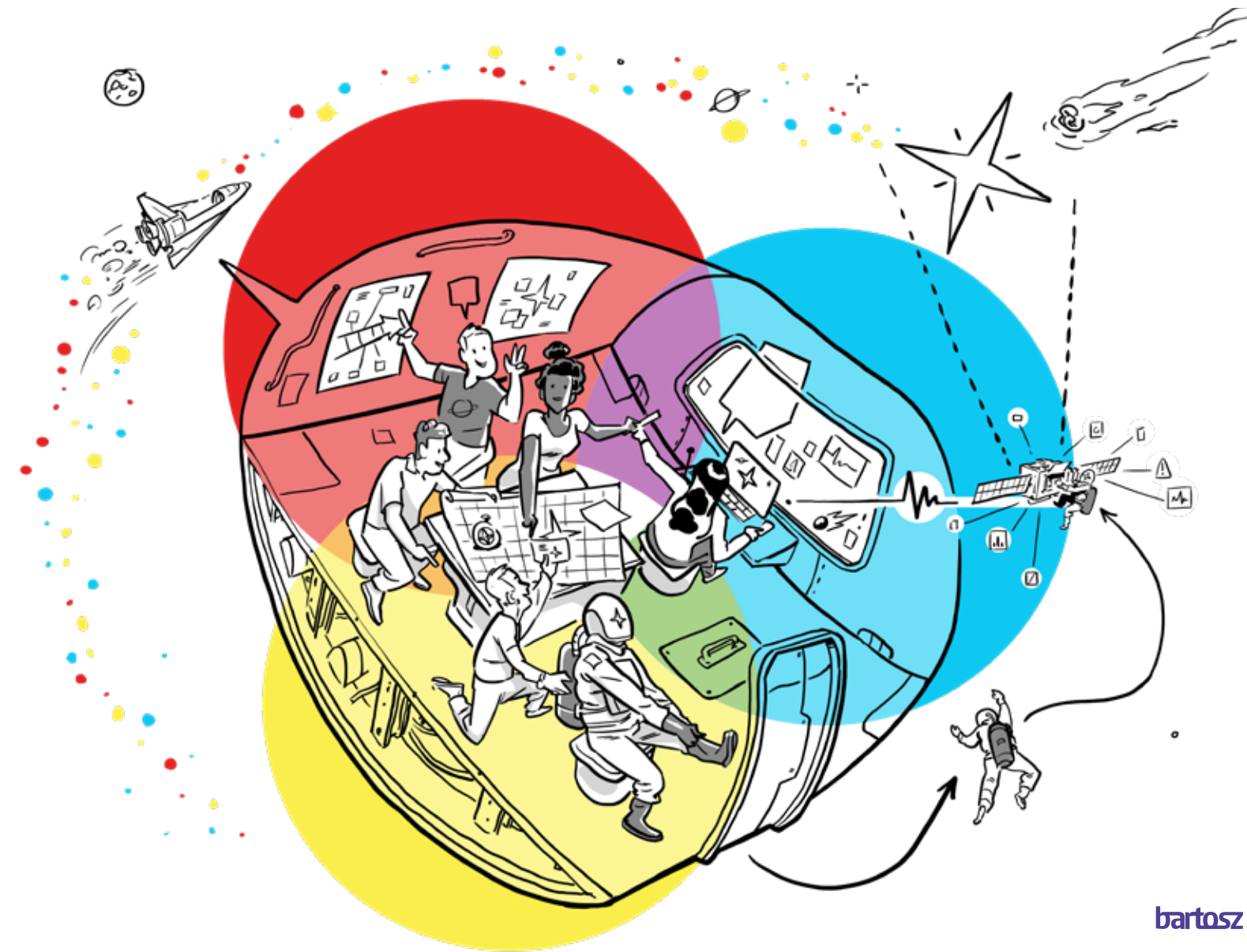
Als je software bouwt is het belangrijk dat er sprake is van een shared understanding: een gezamenlijk beeld van het doel waar het team naar toewerkt. Als er geen shared understanding is, ontstaan er misverstanden en bouw je als team niet wat de klant wil of kan gebruiken. Het hele team voelt zich gezamenlijk verantwoordelijk voor de kwaliteit van het geleverde product, waaraan iedereen vanuit zijn of haar eigen expertise bijdraagt.

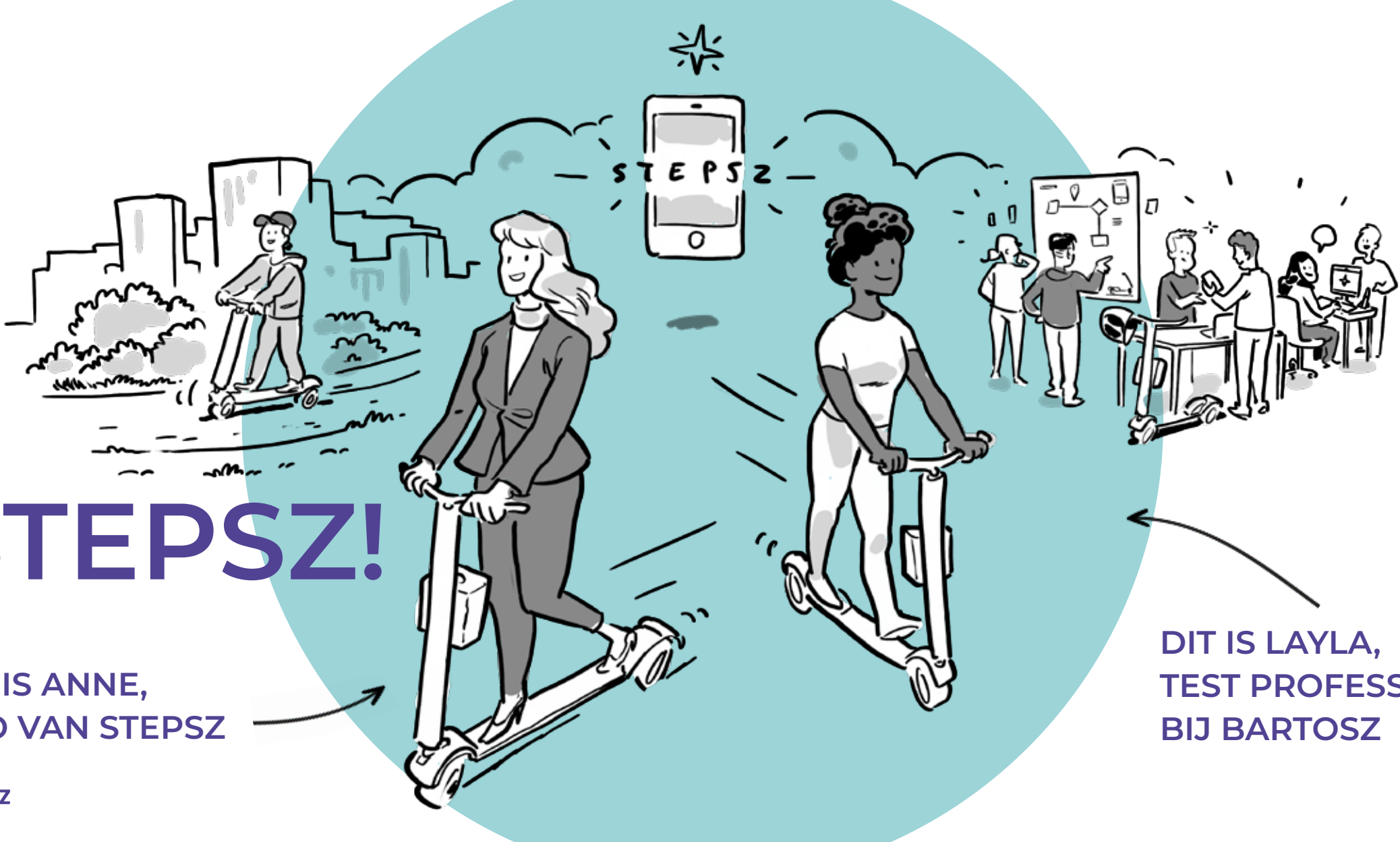
2. FAST FEEDBACK

Hierbij draait het om het krijgen en verwerken van feedback op je product gedurende de volledige software development lifecycle. Software test je tijdens het voortbrengingsproces op allerlei niveaus: van unittest tot ketentest. Bovendien gaat het verder. Je betreft gebruikers in het ontwerpproces, zodat je zeker weet dat je iets bouwt waar zij op zitten te wachten. Inzicht in hoe gebruikers de software gebruiken is waardevolle informatie bij de ontwikkeling van je product.

3. EXPLORATION

Het is onmogelijk om van tevoren alles te weten over je product en met vooraf gedefinieerde tests alle risico's af te dekken. Daarom is het belangrijk om ook te gaan leren en ontdekken, te exploreren. Je leert je software of product kennen door er mee te experimenteren, zodat je niet ontdekte risico's en problemen kunt blootleggen. Ook onderzoek je je proces en de omgeving waarbinnen je opereert.





STEPSZ!

DIT IS ANNE,
CEO VAN STEPSZ

DIT IS LAYLA,
TEST PROFESSIONAL
BIJ BARTOSZ

QIT IN DE PRAKTIJK

EVEN VOORSTELLEN!

Layla werkt aan een opdracht bij Stepsz, een bedrijf dat e-steps (stepsz) exploiteert door heel Nederland. Met een app kun je een step in gebruik nemen, betalen en je route bepalen. Layla werkt als tester en scrum master in het team dat verantwoordelijk is voor het ontwikkelen en onderhouden van deze app.

Stepsz is een klein bedrijf en de lijntjes zijn kort. De CEO en oprichter van het bedrijf, Anne, is ondanks de enorme groei nog steeds erg betrokken bij het product. Het liefst bemoeit ze zich met alles dat binnen het bedrijf gebeurt, maar daar heeft ze nu echt te veel voor op haar bord liggen.

Gelukkig heeft ze in Layla een sparringpartner gevonden die ze vertrouwt. Vanwege haar goede nieuwe ideeën voor de app en vooral als er problemen zijn die opgelost moeten worden.

1.1 ANNES PROBLEEM

Als Layla 's ochtends een kop koffie wil halen, staat Anne haar al op te wachten bij de automaat. Ze ziet van een afstand dat er iets goed mis is. Anne is de vorige dag gebeld door een gemeente waar Stepsz actief is: er wordt bedreigd met het intrekken van de exploitatievergunning. Er komen klachten binnen over foutgeparkeerde stepsz en als dit niet snel opgelost wordt, wil de gemeente de stepsz niet meer in de stad hebben.

Geen paniek: Anne heeft al een oplossing bedacht. Het ontwikkelteam moet 'no parking zones' gaan aangeven in de app. Ze heeft de oplossing ook al besproken met de Product Owner (PO).

Als Layla op de backlog kijkt, dan ziet ze dat de story er al op staat. Na overleg met de analist heeft de PO besloten dat de 'no parking zones' rood moeten worden en de rest van de kaart groen. Layla hoort de ontwikkelaars mopperen: de oplossing klinkt eenvoudig maar is technisch lastig te realiseren en bovendien complex om te onderhouden.





1.2 LAYLA'S ANALYSE

Alle betrokkenen vinden het belangrijk dat de app kwalitatief goed is. Wel heeft iedereen zijn of haar eigen kijk op kwaliteit. Waar de PO en CEO vooral kijken naar waarde voor de klant, letten ontwikkelaars ook op de onderhoudbaarheid en overdraagbaarheid van de software. Daarnaast is voorspelbaar zijn als team in een agile omgeving ook erg belangrijk! De grote story die de PO op de backlog heeft gezet brengt deze aspecten in gevaar, dus het team doet er goed aan om hier kritisch op te zijn. Klakkeloos bouwen wat de PO vraagt is in dit geval veel werk en geeft niet persé het gewenste effect.

We zouden als team de vragen moeten stellen: Wat proberen we te bereiken? Waarom doen we dit? Het antwoord is simpel: we willen minder klachten bij de gemeente, zodat onze vergunning niet ingetrokken wordt. Om dat doel te bereiken zijn er wellicht ook andere oplossingen mogelijk, die misschien wel simpeler te realiseren zijn. Er zou bijvoorbeeld een tutorial voor nieuwe Stepsz-gebruikers ontwikkeld kunnen worden, zodat iedereen beter weet waar de stepsz geparkeerd mogen worden.

Ook die oplossing richt zich echter op de gebruikers van de app, net zoals de oplossing van Anne. Layla vraagt het team een brainstorm te houden over mogelijke andere oplossingen. Daaruit volgt een ander inzicht: kunnen we ons niet beter richten op de klagers? Als we stickers op de stepsz plakken met het telefoonnummer van de klachtenlijn, komen de issues direct bij Stepsz terecht en kan iemand de foutgeparkeerde step verplaatsen. De klachten zijn dan snel afgehandeld en komen niet meer bij de gemeente terecht.

Vanuit QIT gezien sluit de manier om dit probleem op te lossen goed aan bij de eerste pijler: de Whole Team Approach. Hierbij is het belangrijk dat het hele team begrijpt waarom (why) en voor wie (who) we iets doen. De why en de who zijn belangrijk om een shared understanding te creëren van het doel waar je naartoe werkt. Daarmee staan alle neuzen dezelfde kant op, worden misverstanden voorkomen en komen er betere oplossingen uit het team.

1.3 TIPS



TIP 1

Als je het gewenste effect wilt bereiken, is het cruciaal dat je als team weet welke waarde je gaat leveren en voor wie je aan het bouwen bent. Stel je als team dan ook altijd de vraag: begrijpen we de why en de who? Dit creëert een shared understanding van je gezamenlijke doel, wat goed aansluit bij de agile manier van denken. In scrum wordt er een Product Goal opgesteld waaraan increments getoetst kunnen worden om er zeker van te zijn dat die bijdragen aan dit doel. Impact mapping geeft invulling aan het inzichtelijk maken van dit concept.

TIP 2

Gebruik de expertise van het hele team. In het voorbeeld hebben alleen de PO en de analist naar het probleem gekeken. Wanneer ze een ontwikkelaar hadden betrokken, hadden ze meteen geweten dat de oplossing simpel lijkt, maar dat het enorm veel werk is. Om dit toe te passen kun je 'three-amigo-sessies' organiseren. Hierbij zitten teamleden uit de drie disciplines (business, development en test) bij elkaar om functionaliteit uit te werken en kunnen ze hun verschillende inzichten op het gebied van kwaliteit met elkaar delen.



TIP 3

Ten slotte: kent iedereen zijn rol? De PO heeft als enige het mandaat om beslissingen te nemen over alle ontwikkelingen aan het product. Als stakeholder kan de CEO een wens of probleem aangeven bij de PO. Als de PO besluit om hier iets voor te ontwikkelen (de 'wat'), bepaalt het ontwikkelteam vervolgens welke oplossing gebouwd wordt en hoe die oplossing gebouwd wordt (de 'hoe'). Als er sprake is van een shared understanding, kan iedereen de verantwoordelijkheid nemen die bij zijn of haar rol past en op elkaar vertrouwen. De scrum master kan bijvoorbeeld de PO coachen in het managen van de stakeholders.

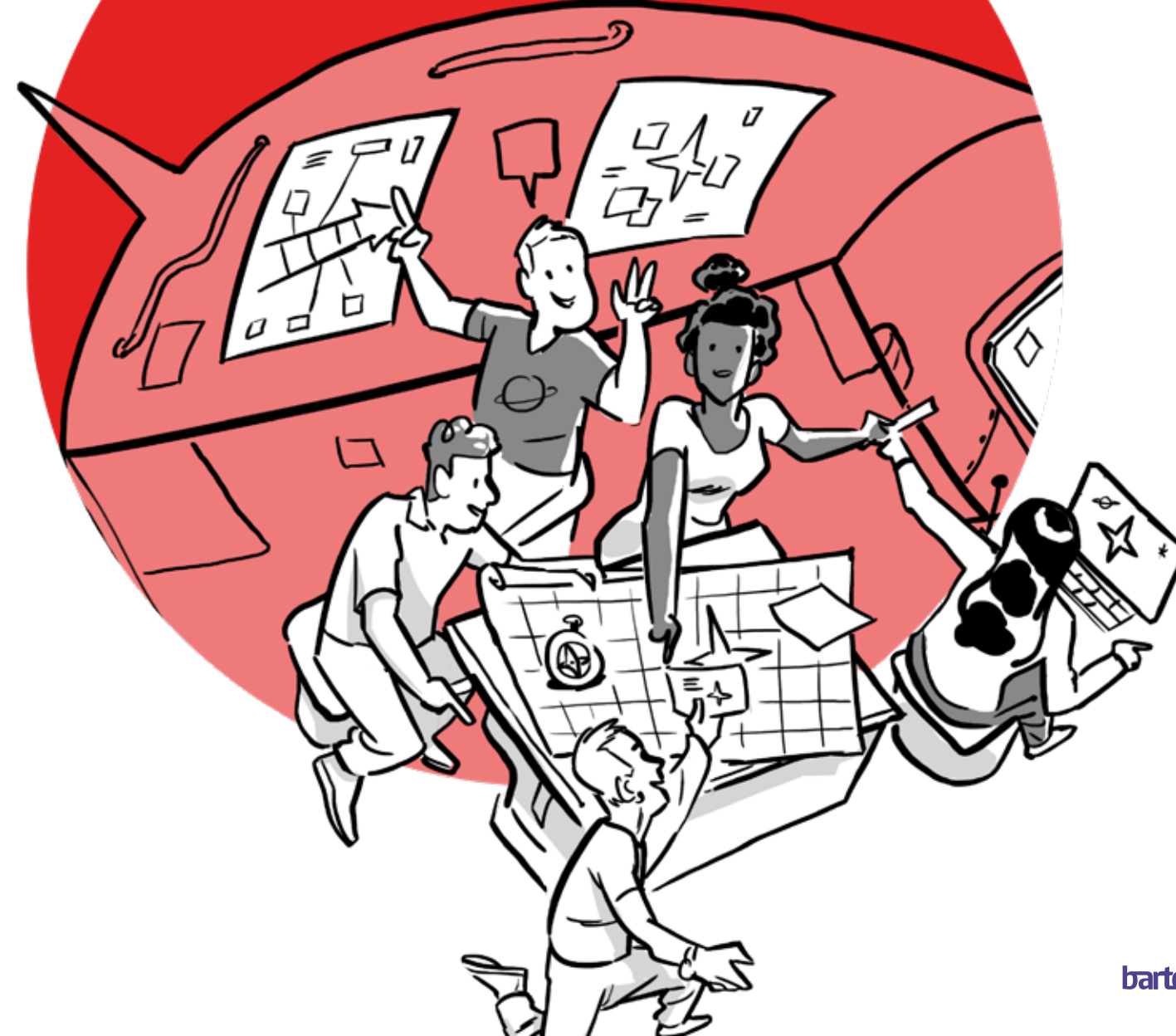
1.4 THEORETISCH KADER

In dit voorbeeld had Anne, als CEO en stakeholder, een probleem. In plaats van dit probleem via de PO op de backlog te plaatsen, om vervolgens het team met een oplossing te laten komen, plaatsten Anne en de PO een oplossing op de backlog. Binnen de Whole Team Approach is het belangrijk dat iedereen binnen het team betrokken wordt bij wat en hoe er gebouwd wordt. Als er een shared understanding is, worden misverstanden voorkomen en voelt iedereen zich verantwoordelijk voor het eindproduct. Als alle teamleden vanuit hun expertise hun mening kunnen geven en kunnen bijdragen, gaat de kwaliteit omhoog!

 **Whole Team Approach:** Het hele team, dus niet alleen de tester, voelt zich verantwoordelijk en kan bijdragen aan de kwaliteit van het product.

 **Belangrijke elementen in Whole Team Approach:** samenwerking, zingeving en doel van het team, informatie visualisatie, communicatie, teamdynamiek, werkprocessen

 **Google suggesties:** teamcharter, liberating structures, Tukman model, invest mnemonic, specification by example

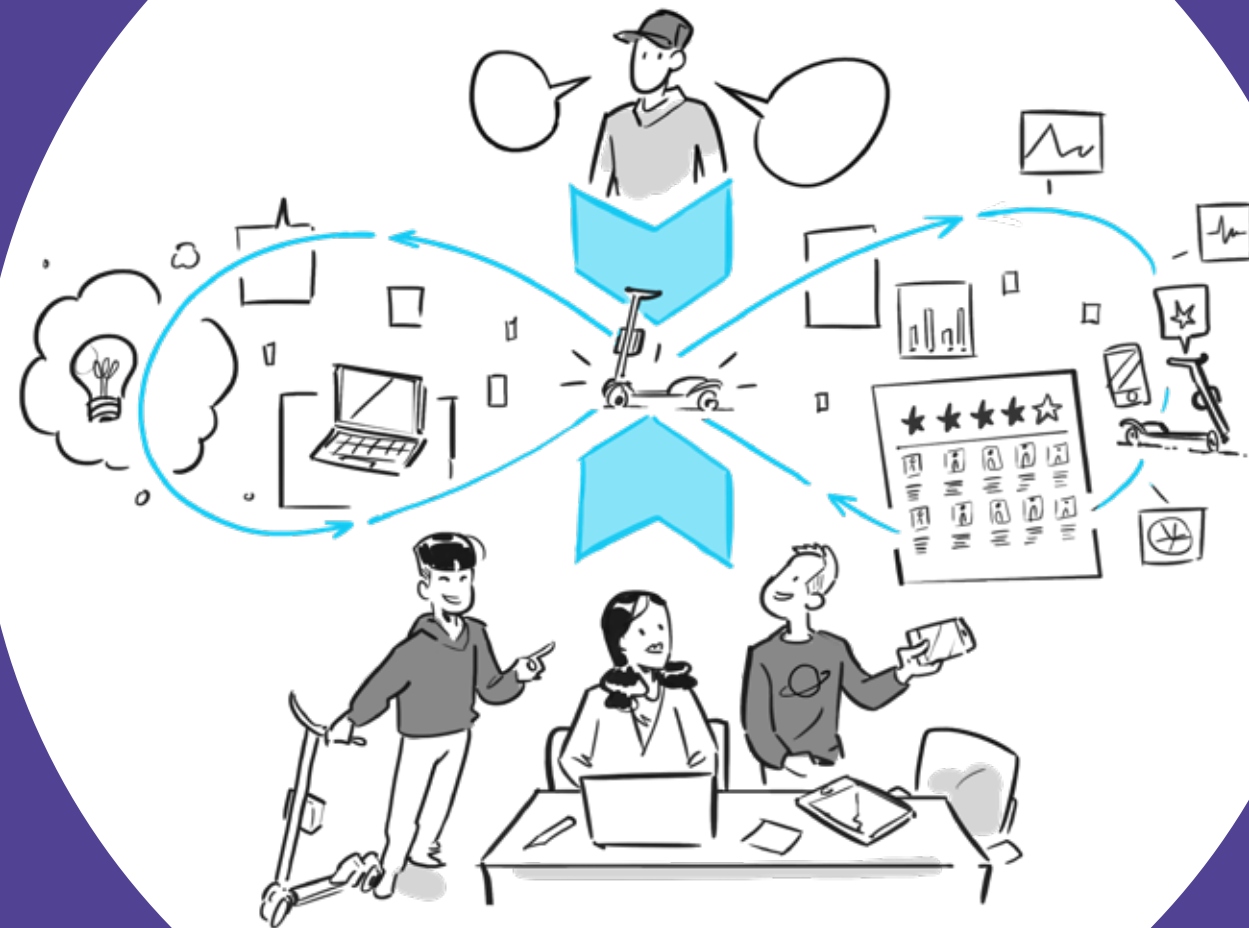


2.1 ANNES PROBLEEM

Tijdens de lunch loopt Layla met haar dienblad door de drukke cafetaria. Het zit vol, maar gelukkig heeft Anne een plekje voor haar vrij gehouden. Layla en haar team hebben net een demo gegeven aan de PO en ze wil niet opscheppen, maar de reacties waren lovend. Jammer genoeg reageert Anne niet zo enthousiast als Layla had gehoopt. Anne verzucht dat de PO dan niet kritisch genoeg is, want er is vast ruimte voor verbetering. De gebruikers vinden dat in ieder geval, anders zouden we in plaats van 4 wel 5 sterren krijgen in de App Store. Het team moet een tandje bijschakelen, want de app is nu niet goed genoeg!

Na de lunch checkt Layla de app in de App Store. De app krijgt 4.2 sterren. Als ze even verder zoekt ziet ze dat vergelijkbare apps waarmee je vervoersmiddelen zoals fietsen of scooters kunt huren nooit 5 sterren krijgen. Vergeleken met andere apps doen ze het helemaal niet slecht. Als Layla wat dieper in de reviews duikt, ziet ze dat gebruikers ook inhoudelijke feedback hebben achtergelaten: klachten en verbeter suggesties die ze nog niet eerder gehoord heeft.





2.2 LAYLA'S ANALYSE

Anne is op zoek naar houvast en inzicht. Ze wil weten of gebruikers tevreden zijn en verbindt daarom de score van de app direct aan de kwaliteit die het ontwikkelteam oplevert. Maar de score wordt door veel meer dingen beïnvloed: het aantal sterren is geen objectieve indicator van kwaliteit. Het is natuurlijk wel belangrijk dat zowel Anne als het ontwikkelteam inzicht hebben in wat de kwaliteit van de app dan wel is en dat is nu niet het geval.

Er wordt bruikbare feedback achtergelaten door gebruikers. Layla vindt het verontrustend dat deze niet bij het ontwikkelteam terechtkomt. De feedbackloop van gebruikerservaringen richting het team ontbreekt nu volledig. Inzicht in wat je gebruikers van je product vinden en hoe ze deze gebruiken is essentieel bij het verbeteren van je product. Stepsz moet het ophalen van feedback in eigen hand gaan nemen.

Op dit moment is het plaatsen van een reactie in de App Store het eerste moment waarop gebruikers hun wensen en frustraties kunnen uiten, wat veel te laat is: de feedback is dan eigenlijk niet meer mee te nemen in het ontwikkelproces. Dat is hetzelfde als je brandmelder als kookwekker gebruiken wanneer je aan het koken bent. Als die afgaat, is je eten al verpest. Het is belangrijk om zo vroeg mogelijk feedback in te winnen zodat je de koers kunt bijstellen en een product kunt maken waar je gebruikers blij mee zijn. Hier draait het om bij Fast Feedback: verzamelen van feedback moet je niet uitstellen, als er iets mis is wil je dat zo snel mogelijk weten. Dit geldt voor gebruikerservaringen, maar ook voor technische issues, of problemen die niet met de app maar met de rest van je product te maken hebben. Op elk niveau wil je zo vroeg mogelijk feedback inwinnen.

2.3 TIPS

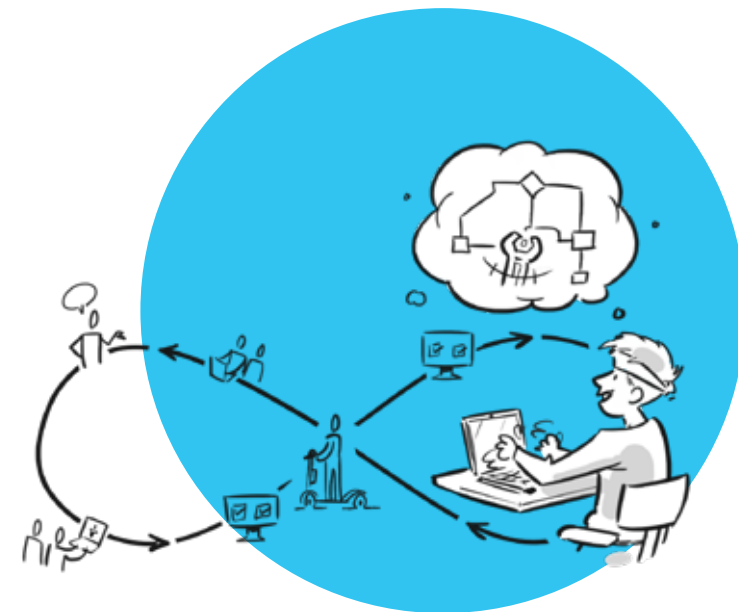


TIP 1

Test je product zo vroeg mogelijk. Dit kun je bijvoorbeeld doen door een 'shift left' te maken. Bij shift-left testing focus je op het zo vroeg mogelijk in het ontwikkelproces feedback ophalen en gebruiken. Betrek bijvoorbeeld de gebruikers bij het opstellen en prioriteren van nieuwe features: zij weten het beste wat je product mist. Als je gebruikers feedback laat geven op wat je wilt gaan ontwikkelen, test je je plannen en weet je eerder of je wel het juiste aan het ontwikkelen bent.

TIP 2

Test je software ook in productie. Dit houdt in dat je testen uitvoert met de software die daadwerkelijk gebruikt wordt. Hierdoor krijg je meteen feedback op wat je ontwikkeld hebt. Je kunt gebruikers vragen om feedback te geven en bugs in te sturen. Maar je kunt ook kijken naar hoe je software gebruikt wordt, zoals welke knoppen en flows het meest gebruikt worden of voor de meeste inkomsten zorgen. Met deze informatie kun je meteen aan de slag om je software te verbeteren. Het verplaatsen van testen naar productie wordt de shift right genoemd, omdat je ook test in een latere fase van het ontwikkelproces.



TIP 3

Verdiep je in CI/CD. In plaats van één grote integratie van code, geschreven over meerdere dagen of zelfs weken, wordt code meerdere keren per dag geïntegreerd. Daardoor blijven integratieproblemen klein en is de impact als ze optreden beperkt. Ze zijn op te lossen op het moment dat de geschreven code nog vers in het geheugen bij de ontwikkelaars zit. Met als bijkomend voordeel dat er nog niet is doorontwikkeld op de foutieve code.

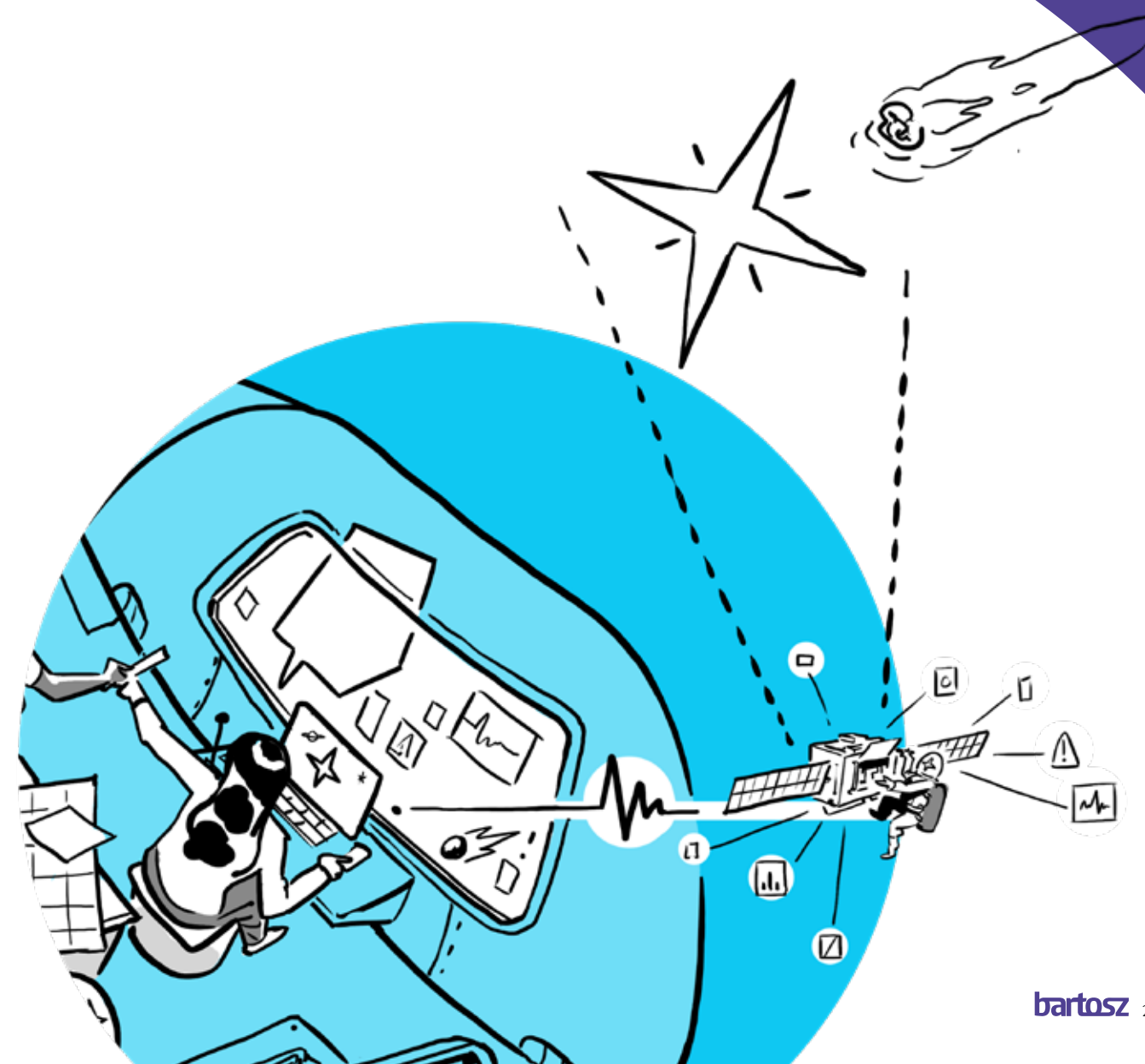
2.4 THEORETISCH KADER

Het voorbeeld van de feedback in de App Store laat zien dat het belangrijk is om inzicht te hebben in je product om de kwaliteit ervan te kunnen verbeteren. Een regressietest die checkt of eerder opgeleverde functionaliteit blijft werken na een nieuwe release is daar een voorbeeld van. Je kunt ook informatie inwinnen over hoe gebruikers omgaan met de software of wat ze graag toegevoegd willen zien worden in de toekomst. Tijdens het hele ontwikkelproces kun je dingen ondernemen om informatie in te winnen, die je kunt gebruiken om beslissingen te nemen die de kwaliteit van je product bevorderen.

 **Fast Feedback:** Het krijgen van de juiste informatie op het juiste moment op basis waarvan het team beslissingen kan nemen en tijdig kan bijsturen.

 **Belangrijke elementen in Fast Feedback:** technische vaardigheden, oog voor detail, (data)analyse, leren van gebruik, bevragen van requirements

 **Google suggestions:** automation in testing, impact mapping, mob programming, a/b testen, testing pyramide



3.1

ANNES PROBLEEM

Aan het einde van de dag geeft Anne een demo aan de wethouder van een stad, waar Stepsz haar diensten wil gaan aanbieden. De wethouder moet groen licht geven voordat Stepsz een exploitatievergunning krijgt. Maar tijdens de demo gaat het helemaal mis: nadat Anne even stil staat om iets uit te leggen, gaat de step niet meer voor- of achteruit. Het lijkt wel alsof de step in parkeermodus staat. Anne krijgt de step met haar account niet meer aan de praat. Gelukkig staat het team stand-by en geeft Layla haar inloggegevens aan Anne. Layla's account werkt wel zodat Anne met de demo door kan gaan.

Voor de wethouder hield Anne het beschaafd, maar als hij vertrokken is laat ze in duidelijke taal merken dat ze absoluut niet blij is met de situatie. Ze snapt niet hoe dit heeft kunnen gebeuren. Alle testen zijn toch geautomatiseerd? Uitzoeken wat er mis is gegaan heeft voor iedereen de hoogste prioriteit.

Het team denkt eerst aan een issue met de step zelf, maar al snel blijkt dat er iets anders aan de hand is. De step kan namelijk ook niet meer ingeleverd worden en is niet meer zichtbaar onder Annes account. Er is wel een andere step in de app te zien, halverwege de demo gehuurd door Annes partner. Het zou niet mogelijk moeten zijn om meer dan één step tegelijk te huren en dit is ook uitvoerig getest. Het team snapt er niets van. Maar dan horen ze Anne mopperen over een belangrijke e-mail die ze gemist heeft omdat ze geen internet op haar telefoon had door een storing bij haar provider. Dat zou er iets mee te maken kunnen hebben!





3.2 LAYLA'S ANALYSE

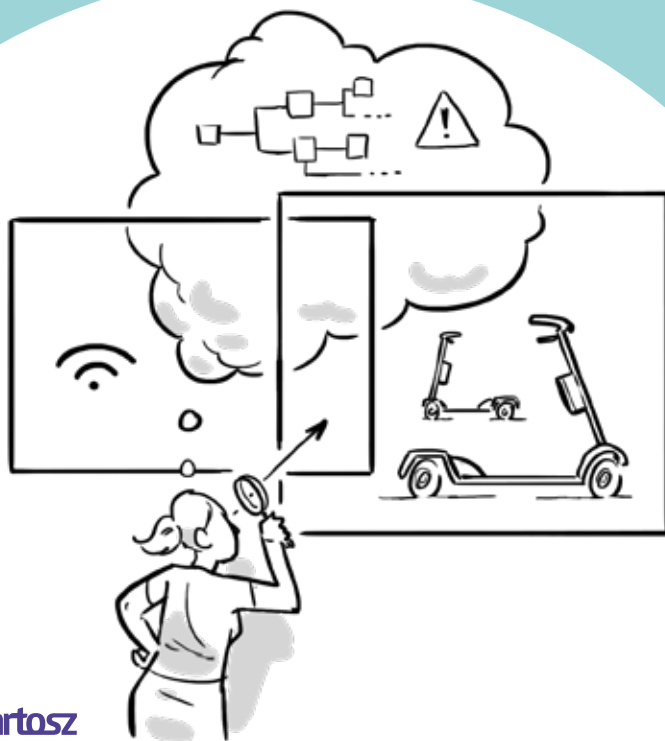
Als onderdeel van de analyse checkt Layla de resultaten van de automatische testen. De test waarin afgedekt wordt dat het niet mogelijk moet zijn om een tweede step te huren als er al één in gebruik is, is geslaagd. Net als de test voor het wegvallen van de internetverbinding. Dan moet de step gewoon door blijven rijden. Vervolgens gaat Layla samen met een ontwikkelaar door de logs die gegenereerd zijn tijdens de demo en dan zien ze wat er gebeurd is. Annes telefoon had geen verbinding met internet op het moment dat er door haar partner nog een step gehuurd werd met hetzelfde account. Als de eerste step geen verbinding maakt met internet, is het mogelijk voor een tweede step om de data te overschrijven en wordt de eerste uit het systeem verwijderd. De eerste step blokkeert dan zodra hij stilstaat en gaat in parkeermodus. De twee issues zijn apart goed getest, maar in combinatie zorgen ze voor een andere uitkomst en blijkt er een probleem op te treden.

De code van de Stepsz-app is ingewikkeld en een product kan altijd onverwacht gedrag vertonen in situaties die je vooraf niet bedacht had. Gebruikers volgen niet altijd de flow die je voor ze uitgestippeld hebt en gebeurtenissen die gecombineerd worden

en opstapelen kunnen onverwachte uitkomsten veroorzaken. Het is daarom belangrijk om je product als geheel te zien, en niet alleen te focussen op de afzonderlijke functionaliteiten. Vertrouw niet blind op je automatisering: Anne sprak het team erop aan dat de fout niet naar voren is gekomen in de geautomatiseerde testen. Deze zijn echter gebaseerd op de input van de tester. Je test daarmee alleen waar de tester aan gedacht heeft. Als alle automatische testen geslaagd zijn betekent dat niet dat je product altijd goed werkt.

Het is belangrijk om je te realiseren dat er altijd onverwachte dingen kunnen gebeuren, ook al heeft het team de Whole Team Approach onder de knie en krijgt het Fast Feedback gedurende het hele ontwikkelproces. Door continu te experimenteren kun je leren over het systeem en de risico's die er zijn. Op basis van de informatie die je opdoet kan de test- en ontwikkelstrategie bijgesteld worden. Accepteer dat je niet alles kunt weten, en dat je niet weet welke informatie je mist. Hoe graag je het ook zou willen. Daarom moet je altijd tijd besteden aan exploreren, aan experimenteren en aan leren over de software. Dat is de basis voor de derde pijler van Quality Infected Teams: Exploration.

3.3 TIPS



TIP 1

Gebruik exploratory testing. Hierbij staan zowel de creativiteit, het echte denkwerk als de nieuwsgierigheid van de tester centraal. De tester combineert deze vaardigheden om zoveel mogelijk te leren over de software die hij of zij aan het testen is. Bij exploratory testing ben je heel bewust bezig met een deel van het product en besluit je in het moment wat de meest zinvolle weg is om meer te ontdekken over het gebied waar je op focust. Omdat je niet op voorhand alle verwachtingen over de werking van het product vastlegt in procedures of testscripts, kun je je strategie steeds bijstellen op basis van wat je al doende leert. Zo ga je dingen ontdekken waar je vooraf geen rekening mee had gehouden.

TIP 2

Wees je bewust van biases die het testen kunnen beïnvloeden.

Bekende voorbeelden zijn de confirmation bias, waarbij je informatie zo interpreteert dat het ondersteunt wat je al gelooft en de automation bias, waarbij je informatie van computers meer vertrouwt dan die van mensen. Deze biases kunnen ervoor zorgen dat het moeilijk is om onbevooroordeeld naar de software te kijken. Het helpt al als je je hiervan bewust bent zodat je ze kunt doorbreken, maar je kunt ook hulp inschakelen. Als je iemand uit een ander team zonder voorkennis jouw software laat testen zou het goed kunnen dat die hele andere dingen opmerkt dan jijzelf.



TIP 3

Reflecteer op hoe, wat en wanneer je test door middel van een test strategy retrospective. Door het doen van zo'n retro wordt duidelijk welke keuzes er gemaakt worden in het testproces en waarom. Als je deze strategie combineert met de risicoanalyse, wordt duidelijk of de juiste focus wordt gelegd en of er risico's zijn die niet afgedekt worden. Het is belangrijk dat het hele team bij de teststrategie en risicoanalyse betrokken is! Zo krijg je inzicht in welke activiteiten uitgevoerd worden door de verschillende teamleden en welke meer aandacht verdienen.

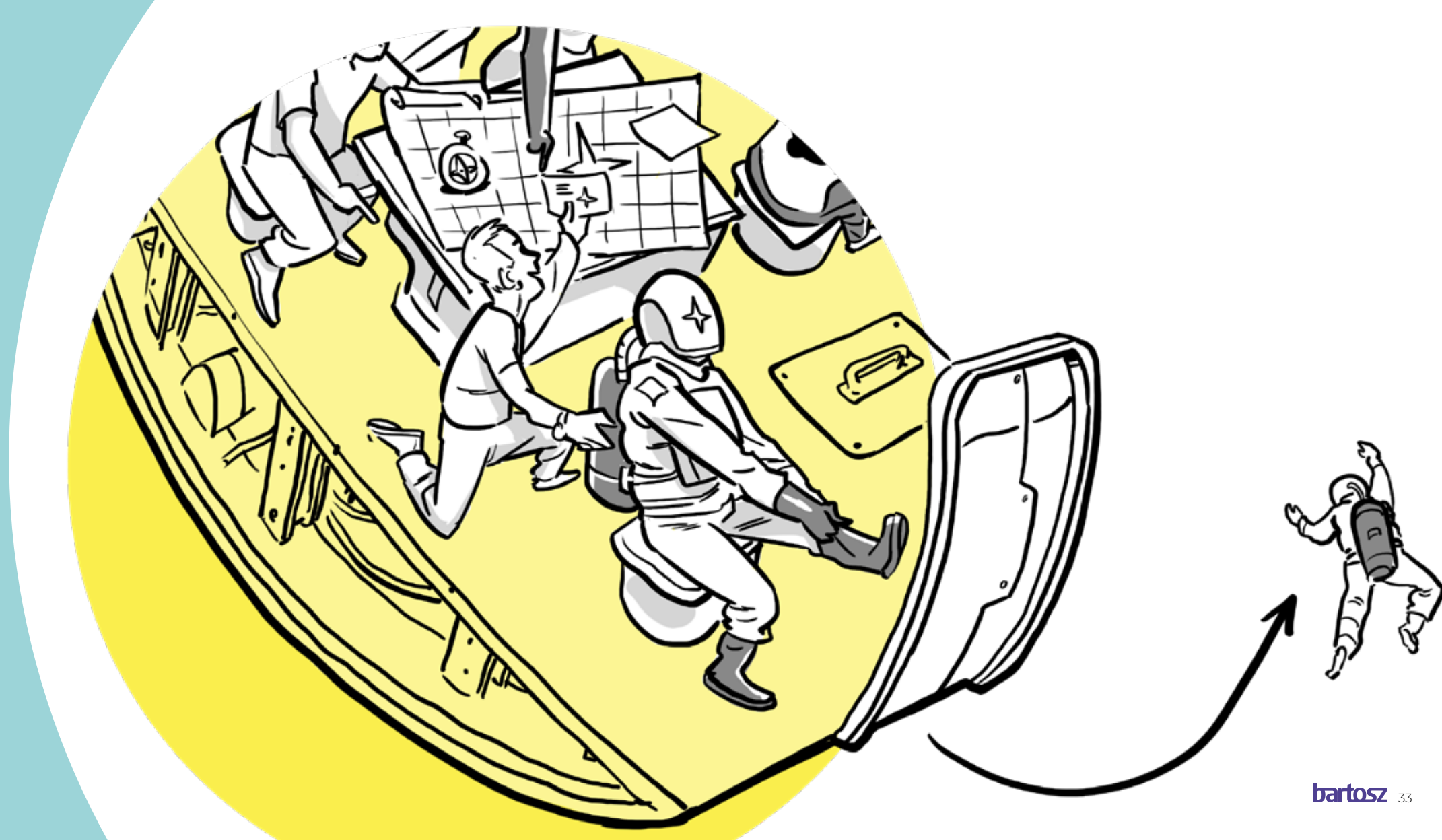
3.4 THEORETISCH KADER

In dit voorbeeld zorgde de combinatie van twee functionaliteiten voor problemen. Dit laat zien dat er altijd onverwachte dingen kunnen gebeuren en dat het onmogelijk is om alles van tevoren te weten en te bedenken. Daarom is het belangrijk om tijd te besteden aan experimenteren met en leren over de software, zodat je ook uitzonderlijke situaties tegenkomt. Als het nodig is kun je je test- en ontwikkelstrategie bijstellen op basis van de inzichten die je hebt opgedaan.

 **Exploration:** Het continu onderzoeken van je product, proces en omgeving en op zoek blijven gaan naar het onbekende, omdat er altijd onverwachte dingen kunnen optreden.

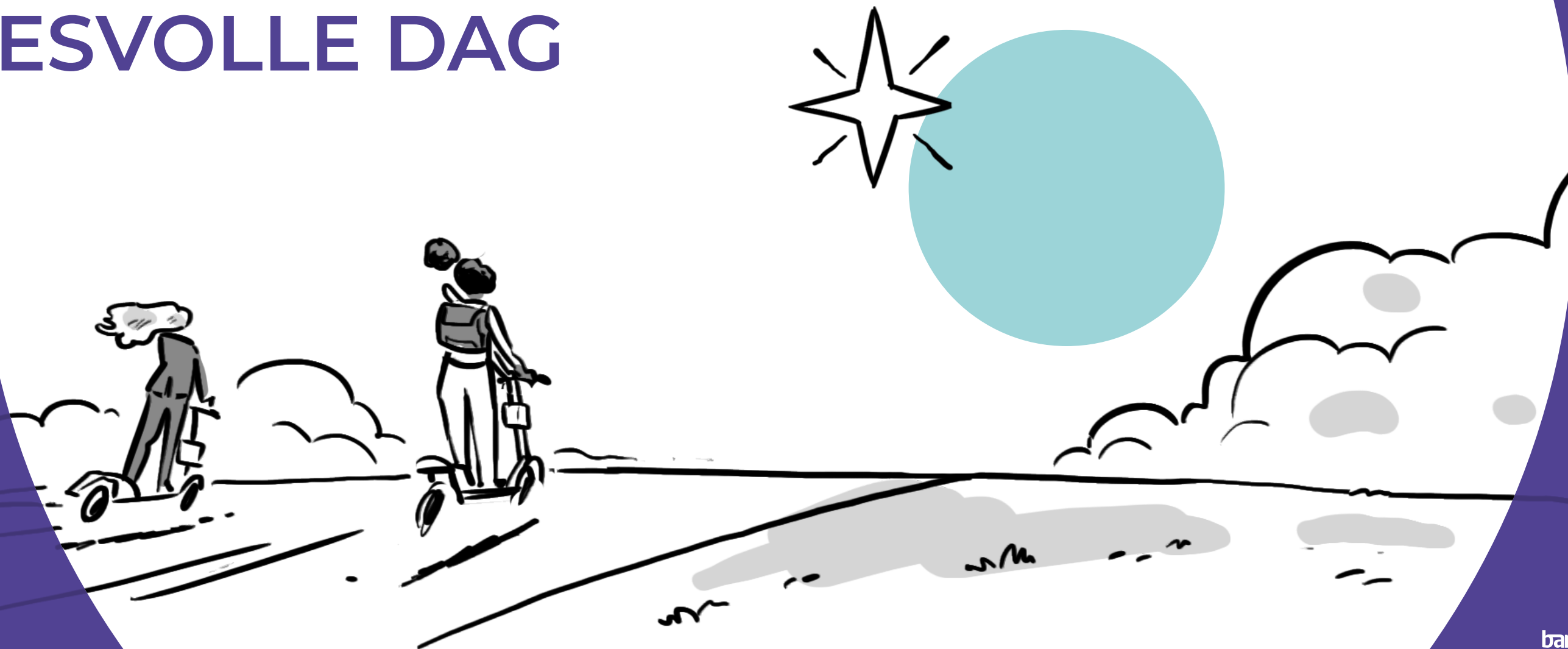
 **Belangrijke elementen in Exploration:** nieuwsgierig, outside the box denken, creativiteit, onderzoeken, psychologie van de tester

 **Google suggesties:** bughunt, rapid software testing, waste snake, critical thinking, testing heuristics



EEN SUCCESVOLLE DAG

Als het tijd is om naar huis te gaan en Layla op de lift staat te wachten, komt Anne aanlopen. Ze ziet er een stuk vrolijker uit en bedankt Layla voor alle hulp. Dankzij haar hebben ze de problemen op kunnen lossen! Layla is blij dat Anne zo enthousiast is over de resultaten die bereikt zijn, maar ze is niet de enige die Anne daarvoor moet bedanken. Layla heeft de richting aangegeven, maar het team heeft met elkaar de keuze voor kwaliteit gemaakt. Als tester heeft Layla de rol van reisleader op zich genomen door iedereen te herinneren aan het belang van focussen op kwaliteit en het team te helpen dit in de praktijk te brengen. We noemen het niet voor niets 'een reisgids voor Quality Infected Teams'. Het blijft een reis, en er zijn altijd volgende stappen te nemen om nog beter te worden.



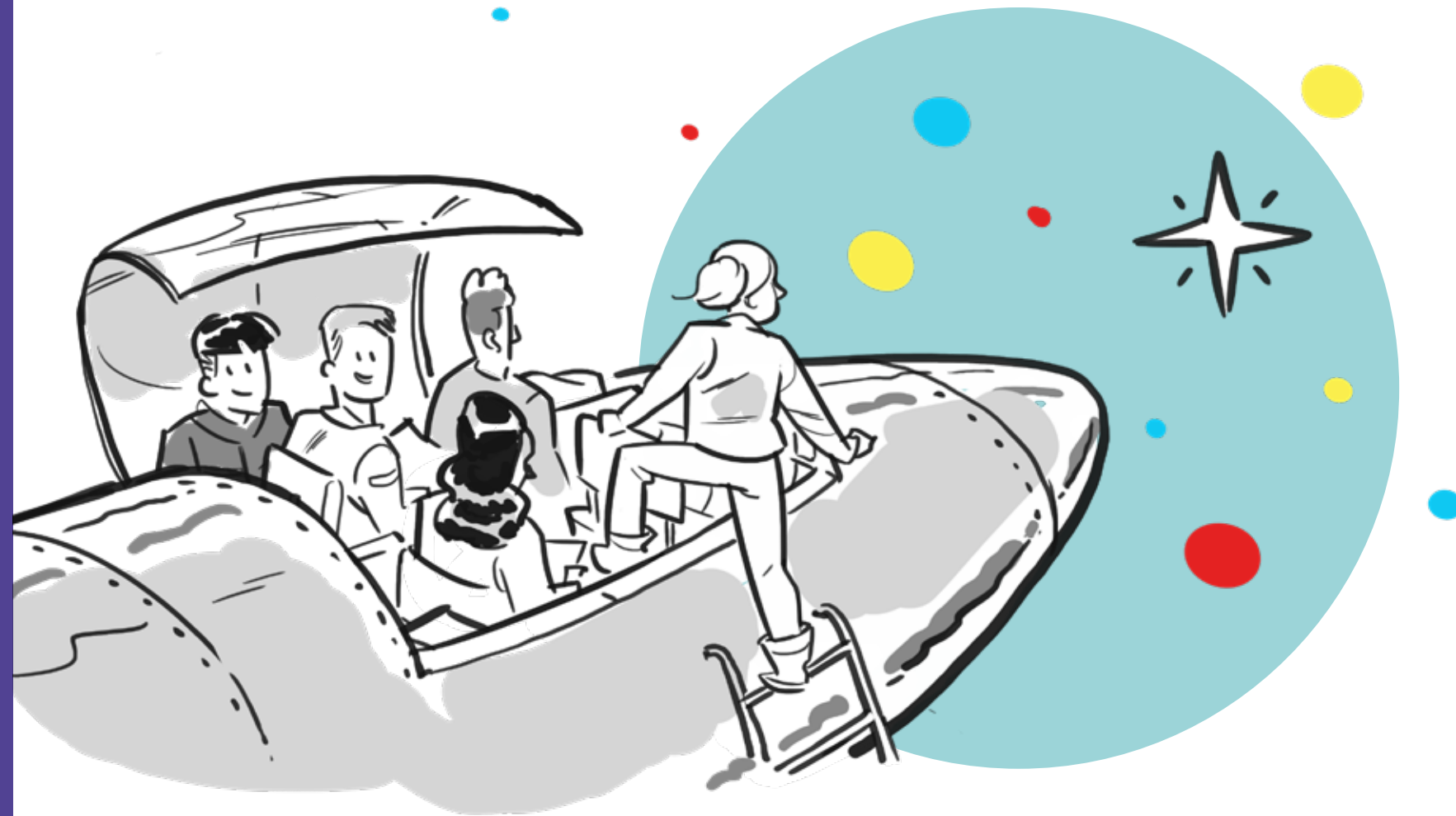
HOE NU VERDER? ...ALS TEAM

Hopelijk heeft dit boekje je interesse gewekt en je aan het denken gezet over welke rol het begrip kwaliteit binnen jouw team speelt. Wij zijn ervan overtuigd dat kwaliteit een expliciete keuze is die gemaakt wordt door het hele team. Kwaliteit is namelijk niet alleen de verantwoordelijkheid van de tester, maar van iedereen. Als je hiermee aan de slag wilt als team, is de eerste stap: reflecteren op de positie die kwaliteit op dit moment inneemt. Bewustwording is altijd nodig om vooruit te komen en het team moet het erover eens worden dat kwaliteit een gezamenlijk doel is. Daar moet een shared understanding over bestaan. Organiseer bijvoorbeeld eens een retrospective over dit onderwerp om erachter te komen wat een expliciete keuze voor kwaliteit zou betekenen voor jouw team.

Van hieruit kun je samen stappen ondernemen om dit verder te onderzoeken. Bartosz organiseert bijvoorbeeld de Quality Infected Teams workshop waarbij je als team gaat onderzoeken

op welke gebieden je wilt focussen. De basis is een kaartspel met conversation starters die verdeeld zijn tussen de drie pijlers: Whole Team Approach, Fast Feedback en Exploration. Tijdens deze workshop worden er kaarten geselecteerd waarop praktische suggesties beschreven staan die het team kunnen helpen om QIT in de praktijk te brengen. Bovendien worden er acties gedefinieerd die meteen in de volgende sprint opgepakt kunnen worden. Neem contact met ons op of vraag een van onze consultants om meer informatie als je geïnteresseerd bent!

Van het kaartspel zijn ook reisvarianten beschikbaar, die je er altijd even bij kunt pakken. Ook wanneer je als team al op de goede weg zit. Je kunt de kaarten gebruiken bij het bepalen van de route naar een steeds beter team! Laat het ons weten als je zo'n kaartspel wilt ontvangen. We sturen graag een set naar je op.

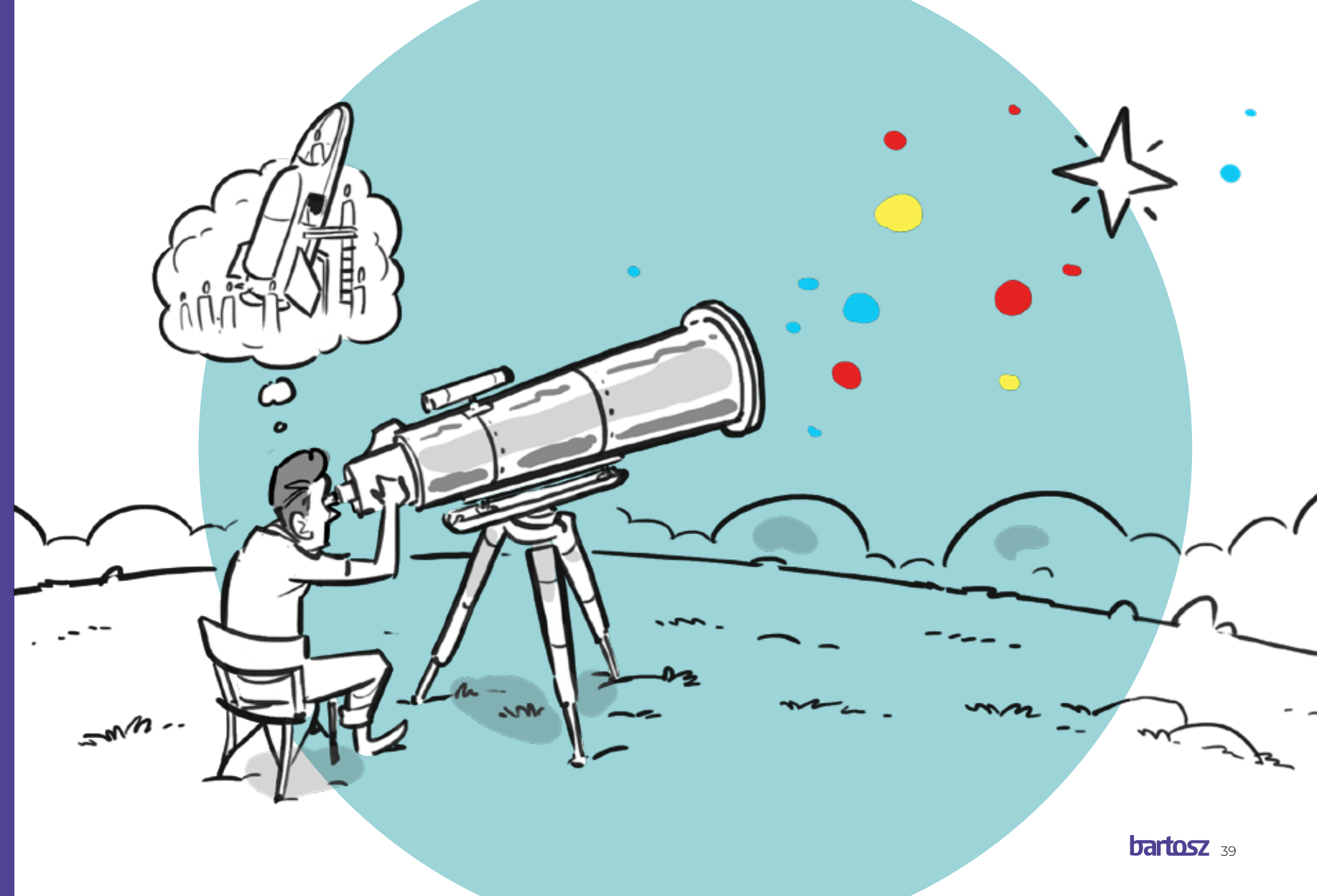


HOE NU VERDER? ...ALS PROFESSIONAL

Dit boekje is niet bedoeld als eindpunt, of als de absolute waarheid. Quality Infected Teams is een visie op kwaliteit binnen de context van agile en scrum. We willen jou als lezer prikkelen om na te denken over je vakgebied en je eigen visie te ontwikkelen op jouw positie binnen een team. We hopen dat dit QIT boekje je aanzet om je eigen verhaal rondom testen en kwaliteit vorm te geven.

Als je je eigen visie duidelijk hebt, kun je van alles doen om jezelf te ontwikkelen tot de professional die daarbij past. Volg trainingen en workshops die je de vaardigheden leren die je kunnen helpen om jouw visie in de praktijk te brengen en je team daarin mee te nemen. Op onze blog op www.bartosz.nl vind je een artikel over ons trainingsmodel dat gebaseerd is op onze QIT visie. Denk ook eens aan het bezoeken van conferenties. Niet alleen om kennis op te doen, maar ook om andere, gelijkgezinde mensen te ontmoeten waarmee je kunt sparren over je ideeën en visie.

Is een conferentie bezoeken nog net een stap te ver, dan biedt Bartosz ook genoeg mogelijkheden om in contact te komen met professionals die kwaliteit en kennis verrijken belangrijk vinden. We organiseren regelmatig sessies waar consultants van Bartosz hun ervaringen en kennis delen en waar je met hen in contact kunt komen. Je bent natuurlijk ook altijd welkom om een keer koffie te komen drinken op een van onze locaties, zodat we je meer kunnen vertellen over wat wij doen en we ideeën uit kunnen wisselen. Stuur ons een berichtje, dan maken we een afspraak!



The image shows a two-page spread from a notebook. The left page has a teal background. At the top, the title "MIJN REISNOTITIES" is written in a white, hand-drawn, uppercase font. Below the title, there is a line drawing of two people riding scooters on a path. One person is in the foreground, and another is slightly behind them. A simple star with radiating lines is drawn above the scooters. In the bottom left corner, the word "bartosz" is written in a small, white, lowercase font. The right page is white and features horizontal blue lines for writing. In the bottom right corner, the word "bartosz" is written in a small, blue, lowercase font. The entire notebook is framed by a teal border.





TOT SLOT

Het beste verbeteren begint vandaag! In dit 'Improve Your Best Team' boekje staat onze visie op kwaliteit beschreven, met als doel om de teams waarin wij meedraaien nog beter te maken. Bartosz gelooft in het delen van kennis. We hopen jou met dit boekje te prikkelen om jezelf continu te verbeteren. Verbeter jezelf, je team en de kwaliteit van je product door met Quality Infected Teams aan de slag te gaan. Ons doel is bereikt wanneer je geïnspireerd bent geraakt en ideeën hebt opgedaan!

Laat ons weten wat je van dit boekje vindt. Heb je tips, suggesties, complimenten of kritiek? We horen het graag, bijvoorbeeld via info@bartosz.nl. Want we geloven in de kracht van feedback!

Alle personages, gebeurtenissen en entiteiten in dit boekje zijn fictief. Elke overeenkomst met bestaande personen, gebeurtenissen of entiteiten berust op louter toeval.

Dit boekje is gemaakt door Bartoszians en voortgekomen uit een Bartosz Labsz experiment. Wil je op de hoogte blijven van innovaties door Bartosz? Volg ons dan via social media en neem een kijkje op de Bartosz blog: www.bartosz.nl/blog.

IMPROVE YOUR BEST!

Volgens Anne, de CEO van Stepsz, zijn er drie grote problemen met de kwaliteit van de app die het bedrijf maakt. Layla, tester en scrum master, probeert deze op te lossen met de Quality Infected Teams visie in haar achterhoofd en komt er al snel achter dat elk probleem te maken heeft met één van de drie pijlers van QIT.

#BARTOSZICT



BARTOSZ ICT | T 088 227 86 79 | WWW.BARTOSZ.NL
CITADEL 8 | 3905 NK | VEENENDAAL
Oude Middenweg 17 | 2491 AC | DEN HAAG