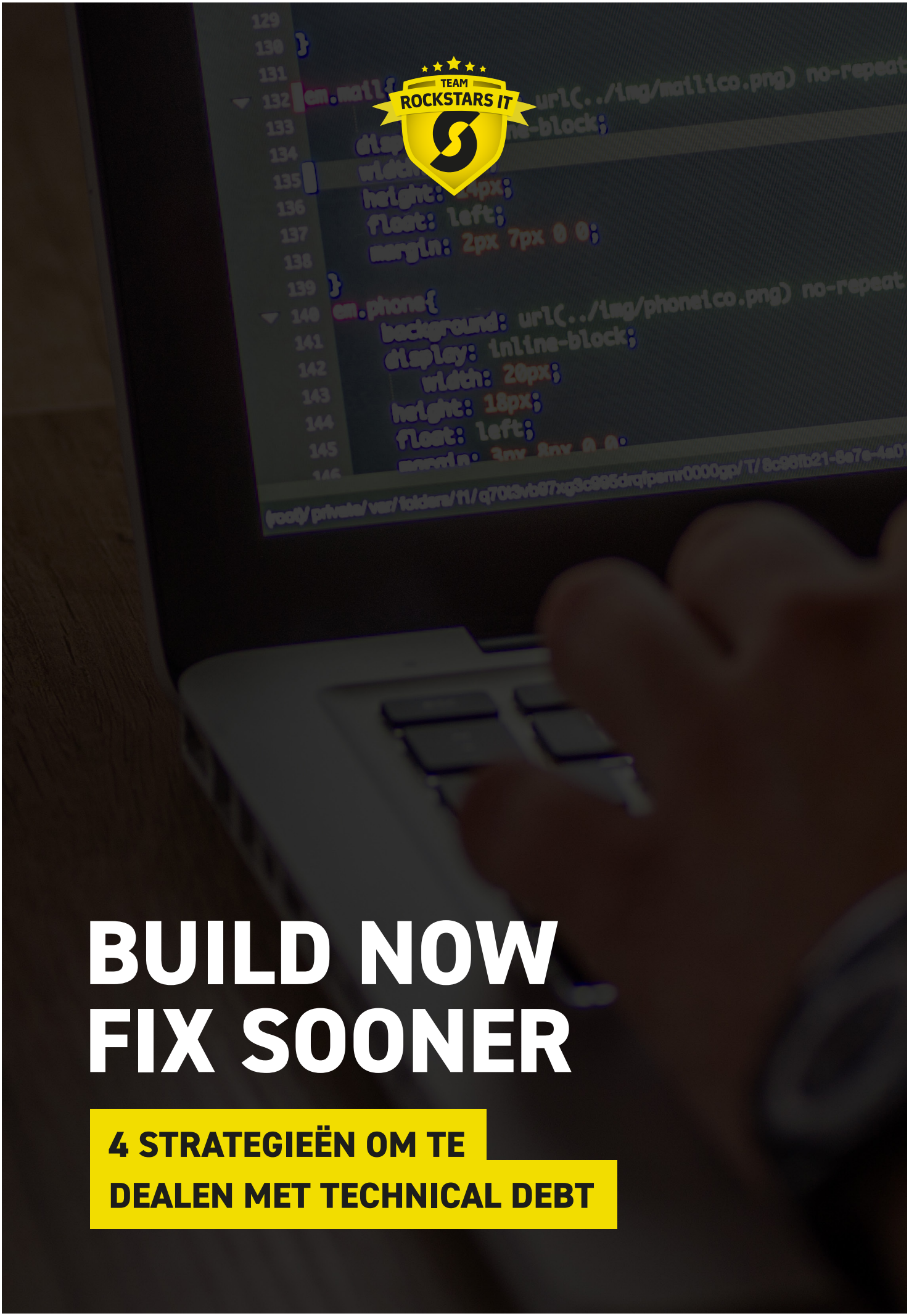

BUILD NOW FIX SOONER

4 STRATEGIEËN OM TE DEALEN MET TECHNICAL DEBT

BUILD NOW, **FIX SOONER**

4 STRATEGIEËN OM TE DEALEN MET TECHNICAL DEBT

Je kent het wel. Een app die gisteren gelanceerd had moeten worden, een nieuwe feature die vanmorgen al had moeten werken: software development vraagt soms om quick & dirty fixes om deadlines te halen. Deze 'build now, fix later' mentaliteit kan op zich geen kwaad, maar als de snelle, ondoordachte oplossingen zich blijven opstapelen ontstaat er technical debt. Technische schuld waar rente over wordt betaald in de vorm van verminderde productiviteit, onbeheersbare code en grotere risico's op bugs. Hoe ga je om met technical debt? Is er voldoende begrip vanuit de business en zo niet, hoe krijg je dat begrip?

WAT IS TECHNICAL DEBT?

Bij softwareontwikkeling ontstaat technical debt wanneer ontwikkelaars kiezen voor suboptimale oplossingen om deadlines te halen of resultaten te leveren. Denk dan aan het schrijven van code die moeilijk te onderhouden of te begrijpen is, het niet volgen van best practices en het uitstellen van noodzakelijke refactoring of het opschonen van code. Technical debt stapelt zich op tot het bijna onmogelijk is om wijzigingen aan te brengen, nieuwe functies toe te voegen of problemen op te lossen. Hoe langer technical debt zich opstapelt, hoe moeilijker het wordt om ervan af te komen. Om te zorgen dat de technical debt behapbaar blijft, moeten developers streven naar een balans tussen het behalen van korte termijn doelen en het in stand houden van de gezondheid van de software. Dit kan bijvoorbeeld door het periodiek inruimen van tijd en middelen om te refactoren en code op te schonen.

HET SPANNINGSVELD TUSSEN IT EN 'DE BUSINESS'

Technical debt is over het algemeen het resultaat van het spanningsveld tussen 'de Business' die continu en snel wil innoveren en IT die dat op een gecontroleerde en juiste manier wil doen. In een tijd waarin IT steeds meer onderdeel wordt van de core business wordt het belang van het reduceren van technical debt gelukkig steeds duidelijker, maar dat vlakt de behoefte aan innovatie vanuit de business niet uit. Het blijft zoeken naar de optimale balans, want te veel technical debt resulteert uiteindelijk in een business die niet optimaal functioneert (en wellicht zelfs tijdelijk geen nieuwe features kan uitbrengen). Een nauwe samenwerking en gedeelde verantwoordelijkheid is hierbij de oplossing.

Een effectieve samenwerking tussen business en IT begint met het creëren van een cultuur van open communicatie en wederzijds begrip. Beide partijen moeten elkaars doelen, uitdagingen en beperkingen begrijpen om gezamenlijk tot optimale oplossingen te komen. Door regelmatig overleg en een gestroomlijnde besluitvormingsstructuur kunnen de business en IT snel reageren op veranderende behoeften en nieuwe kansen voor innovatie identificeren.

Daarnaast maak je de business en de IT idealiter samen verantwoordelijk voor technical debt. Het moet niet zo zijn dat technical debt eerst tijdenlang opstapelt en dat begrip vanuit de business pas ontstaat op het moment dat het eigenlijk al te laat is. Gedeelde verantwoordelijkheid voor het verminderen of controleren van technical debt kan een cultuur van duurzame technische groei bevorderen. Er ontstaat zo een mindset die gericht is op kwaliteit, stabiliteit en onderhoud.

HOE GA JE OM MET TECHNICAL DEBT?

Om te zorgen dat technical debt beheersbaar blijft, kunnen verschillende strategieën ingezet worden. We zetten een aantal van deze strategieën op een rij:

1. VRIJE SPRINTS DOEN

Reserveer periodieke sprints om specifiek en structureel aan technical debt te werken. Dit helpt om de technical debt geleidelijk aan te pakken en de codebase te verbeteren zonder het hele systeem te herbouwen. Bovendien geef je je developers op deze manier autonomie en verantwoordelijkheid voor het op orde brengen van technical debt.

2. VOLLEDIG REFACTOREN

Naast vrije sprints is er ook de mogelijkheid om voor een langere tijd alle features stil te leggen en de technical debt van bijvoorbeeld het afgelopen jaar in één keer op te lossen. Hierbij bestaat wel de kans op over-engineering, waarbij erg veel tijd gestoken wordt in een klein stukje software dat relatief weinig belang heeft. Het is belangrijk om te zorgen dat de software die aangepakt wordt ook voldoende relevantie heeft. Hou er verder rekening mee dat het volledig refactoren van een systeem kan leiden tot een spanningsveld ten opzichte van de business omdat er tijdelijk geen nieuwe functies worden opgeleverd.

3. MAAK GEBRUIK VAN DE VOORDELEN VAN DE CLOUD

Gebruik bijvoorbeeld externe bouwblokken om bestaande componenten (met veel technical debt) te vervangen. Deze bouwblokken zijn op de juiste manier gebouwd en kunnen soms zelfs zonder enige aanpassing ingezet worden. Zo los je veel technical debt in één keer af. Het risico is echter dat er een Cloud lock-in ontstaat, en zulke componenten hebben uiteraard een prijskaartje.

4. TECHNICAL DEBT ACCEPTEREN

Erken dat technische schuld onvermijdelijk is en calculeer in dat er soms bugs kunnen ontstaan. Het is niet nodig om kosten wat het kost voor 99% security en kwaliteit te gaan. Accepteer dat 80% voldoende is en zorg dat je een goede strategie hebt om snel in te spelen op die mogelijke bugs.

VIJF UITSMIJTERS; DOE ER JE VOORDEEL MEE!

1. Breng de kosten en opbrengsten per IT-product in kaart. Investeer in het verminderen van technical debt van IT-producten die de meeste waarde opleveren. Bepaal daarnaast altijd of het opruimen van technical debt meer oplevert dan het maken van nieuwe features. Zo worden altijd goed onderbouwde keuzes gemaakt.

2. Moedig je teams aan om clean code te blijven schrijven, ook als er snel functies geleverd moeten worden. Een goede balans tussen snelheid en codekwaliteit helpt bij het voorkomen van technical debt op lange termijn.

3. Creëer multidisciplinaire teams. Deze teams kunnen technical debt effectiever beheren omdat ze een gedeelde verantwoordelijkheid hebben en daarnaast diverse perspectieven en vaardigheden combineren om tot de beste oplossingen te komen.

4. Stem doelstellingen en KPI's voor business en IT op elkaar af. Dit helpt bij het prioriteren van technical debt die direct van invloed is op de klantwaarde en bedrijfsdoelstellingen. Het zorgt ervoor dat technical debt wordt aangepakt met een focus op het maximaliseren van de waarde voor de gebruikers.

5. Stimuleer open communicatie en samenwerking tussen de business en IT. Moedig regelmatige interactie en kennisuitwisseling aan om een beter begrip te krijgen van elkaars uitdagingen en behoeften. Dit helpt bij het identificeren van potentiële technical debt en het vinden van gezamenlijke oplossingen die de belangen van beide partijen dienen.

Wil je verder sparren over technical debt en mogelijke oplossingen? Neem dan contact op met Team Rockstars IT!