

# To code or not to code

That's the question

# Voorwoord

Bedrijven zijn op zoek naar een slimme manier om aan de stijgende vraag naar software oplossingen te voldoen. De context kan verschillend zijn:

- Keuze voor een nieuwe ERP? Heb ik eigenlijk een ERP nodig?
- Welk softwarepakket moet ik aanschaffen? Of moet ik het zelf bouwen?
- Met welke technologie kan ik eventueel maatwerk oplossingen bouwen?

Beslissingsnemers worden beïnvloed op vele manieren. Een directeur heeft een mooi ogende tool gezien: "Kijk eens naar Monday.com!" Of software is gratis beschikbaar binnen het bedrijf: "We hebben al Microsoft Office 365, dus we kunnen Power BI of Powerapps zonder kosten inzetten." Slimme marketing, want de makkelijkste manier om meer te verkopen, is immers extra verkopen bij bestaande klanten.

Hoe relevant is mooi? En is gratis altijd gratis?

## A Fool With A Tool Is Still A Fool

Met dit document bieden we graag een handvat om tot goede beslissingen te komen rond het implementeren van digitale oplossingen.

De keuze voor technologie is inderdaad een moeilijke en belangrijke keuze. Maar het is niet de eerste vraag die men zich moet stellen, en derhalve van ondergeschikt belang.

In dit document spreken we over 'no-code' en 'low-code'. Het verschil tussen beiden zit in de laagdrempeligheid voor niet-technische profielen om met de tool aan de slag te gaan. We maken verder geen onderscheid tussen de platformen en spreken daarom globaal over 'no-code'.

# Stappenplan voor digitale innovatie

## Begin bij het begin

De kunst bestaat erin om een software architectuur op te zetten die zowel robuust als wendbaar is. Hierbij worden twee invalshoeken bekeken:

Er zijn bedrijfsprocessen die weinig aan verandering onderhevig zijn, of die klassiek binnen zowat alle bedrijven bestaan: denk aan inkoop, verkoop, facturatie, klantenbeheer, etc. Dit soort processen worden meestal goed ondersteund door bestaande softwarepakketten (ERP, CRM, WMS, ...).

Anderzijds zijn er processen die juist vragen om flexibiliteit, of die onderscheidend zijn voor jouw organisatie.

De doelstelling moet zijn om tot een software architectuur te komen, waarbij tools ondergeschikt zijn aan de bedrijfsprocessen en doorheen de tijd kunnen vervangen worden door andere tools.

Lees meer: [Het grote belang van een Bimodal IT architectuur](#)

Hierdoor vermijd je 'vendor lock in', en behoud je de mogelijkheid om te innoveren waar nodig.

## De processen zijn helder: wat nu

Wanneer het duidelijk is hoe de bedrijfsprocessen er uit zien, dien je per proces uit te maken of ze typisch bij jouw basis processen behoren, dan wel, of ze eerder onderscheidend zijn. Die classificatie zal je alvast helpen om ruwweg het onderscheid te maken rond 'software kopen' of 'software bouwen', omdat de meeste basisprocessen vaak identiek zijn binnen een sector, en hier dus vaak al standaard software voor bestaat.

Wanneer je processen helder zijn, zal dit je helpen om in te schatten welke software op de markt voor 80% matcht met jouw noden. Het aanpassen van jouw werkwijze aan de werking van een bestaand softwarepakket valt zeker ook te overwegen (liever dan de software te laten aanpassen).

## Ik vind de ideale oplossing niet!

Nu wordt het duidelijk: er is maatwerk nodig. Maar hoe ga ik hier mee om? Er is een heel duidelijke trend om met platformen te gaan werken die je toelaten om sneller apps te bouwen (Rapid Application Development), of applicaties aan te passen aan wijzigende omstandigheden.

Vooraf voor je onderscheidende processen is dit vaak relevant, want hoe sneller je kan inspelen op veranderende marktomstandigheden, hoe beter. Dit is hét moment om een technologische evaluatie te maken.

Allerlei leveranciers proberen 'hun klant' moreel te verplichten om hun platform te gebruiken door development platformen te bundelen met andere software suites: denk aan Microsoft Office 365, Zoho One, enz. De perceptie dat de gebruikerslicenties gratis zijn en in de kast beschikbaar liggen, creëert voor beslissingsnemers een druk om met die technologie aan de slag te gaan. Mogelijk is dat verstandig, mogelijk ook niet: dat hangt opnieuw af van wat je noden zijn. Het algemene advies is om naast het bekijken van de techniek (hoe kan ik snel applicaties bouwen, welke kennis heb ik nodig, hoe complex is het, etc), ook een goede studie te doen van de verborgen kosten. Zo is ook hier 'gratis' vaak heel relatief. Wat als je koppelt met data buiten de suite? Wat als je externe gebruikers wenst toe te laten op je applicaties? Etc.

# De omgekeerde wereld

De ideale wereld bestaat niet. Dus word je als organisatie soms gedwongen om snelle tactische beslissingen te nemen. Hieronder bespreken we een aantal typische omstandigheden, waarbij het verstrekken vanuit de business processen niet altijd realistisch is. Soms moet een probleem opgelost worden, zonder de hele architectuur in vraag te stellen.

## Shadow IT

Wanneer een bedrijf worstelt met een massa aan Excel, Microsoft Access of andere huis-, tuin - en keukenoplossingen en die problematiek wenst aan te pakken, is een snelle migratie naar een schaalbare applicatie een goede korte termijn aanpak. No-code platformen zijn vaak makkelijke omgevingen om dit te realiseren.

Lees ook: [Hoe los ik mijn Excel nachtmerrie op](#)

## Citizen Development

Wanneer je als bedrijf vooral een oplossing nodig hebt om business gebruikers te betrekken bij het bedenken van nieuwe oplossingen, of bij het faciliteren van het samenwerken van de business gebruikers en ICT, of wanneer je een omgeving kan gebruiken om ideeën af te toetsen aan hun haalbaarheid (MVP), dan zijn no-code platformen hier uitermate geschikt voor. Ze hebben de mogelijkheid om flexibiliteit aan te bieden voor de business (en zo te vermijden dat men zijn toevlucht zoekt naar Excel), terwijl ICT toezicht kan houden op wie, wat, met welke data kan doen (governance & security van gegevens).

# Vergelijking van een paar no- en low-code platformen

We zetten een aantal platformen naast elkaar:

	Betty Blocks	Power Apps	Zoho Creator
Prijspolitiiek	Prijs kan flexibel afgesproken worden: op basis van aantal componenten, apps of users	Complex per-user model. Meerdere MS apps nodig. Gratis binnen O365. Uitbreiding (bv. externe users) leidt tot aanzienlijke verhoging van de kosten	'Per user' model Aparte (lagere) tarieven voor externe gebruikers
Database	Inbegrepen	Nood aan third-party data bronnen	Inbegrepen
Mobiele apps	Een app kan op desktop of mobiel gebruikt worden	Men dient vooraf bij het creëren van een app een keuze te maken voor 'mobile'. Geen PWA ondersteuning.	Een app kan op desktop of mobiel gebruikt worden. Toegang tot de mobiele versie, via de Zoho Creator app. Automatisch genereren van PWA versie
User Interface	Uitgebreide UI/UX mogelijkheden	Niet altijd even gemakkelijk en kan een tijdrovende klus zijn	Beperkte UI/UX mogelijkheden
Citizen Development	★★★	★★	★★
No-code/ Low-code	No-code	Low-code	Low-code
Uitbreiding met code	Javascript React	Steile leercurve om Microsoft Power FX te leren	Deluge is vrij makkelijk te leren

# To code or not to code: FAQ

## Wat zijn de grootste beperkingen van no-code systemen?

Hoewel no-code platformen veel voordelen bieden, is het essentieel om bepaalde beperkingen af te wegen tegen je specifieke projectvereisten en langetermijndoelen. Soms kan een combinatie van no-code en custom code-oplossingen nodig zijn om deze beperkingen te overwinnen en complexere, schaalbare applicaties te bouwen. We lijsten een paar onderwerpen op die je zeker in overweging moet nemen:

- **Mogelijke prestatieproblemen:** Hoewel no-code platformen in veel gevallen goed presteren, kunnen ze beperkingen hebben op het gebied van schaalbaarheid en prestaties. Bij toepassingen met grote hoeveelheden gegevens of complexe logica kan de prestatie mogelijk niet aan de verwachtingen voldoen.
- **Geavanceerde gebruikerservaring:** Als je een zeer gepolijste en op maat gemaakte gebruikerservaring wilt creëren met geavanceerde animaties, interactieve elementen of complexe gebruikersinterfaces, kan een no-code platform beperkt zijn. Het kan nodig zijn om op maat gemaakte code te schrijven om de gewenste functionaliteit te bereiken.
- **Geavanceerde data transformaties en berekeningen:** Als je applicatie vereist dat je geavanceerde data transformaties moet uitvoeren, complexe berekeningen moet maken of gegevens moet analyseren op een manier die buiten de mogelijkheden van de ingebouwde functionaliteit van het no-code platform valt, kan het nodig zijn om aangepaste code te schrijven of een andere ontwikkelingsbenadering te overwegen. De meeste platformen bieden wel de mogelijkheid om met code de functionaliteit uit te breiden.
- **Rapportering is niet altijd een 'core' functionaliteit.** Op zich is hier niets mis mee. Hier bestaan slimme oplossingen voor in de markt (BI mogelijkheden). De meeste platformen maken het eenvoudig mogelijk om data te ontsluiten voor BI tools via een API connectie.

# Wat is het belangrijkste gevaar bij gebruik van Excel?

Het belangrijkste gevaar bij het gebruik van spreadsheets is het risico op fouten en onnauwkeurigheden in de gegevens en berekeningen. Excel is een krachtige tool voor gegevensanalyse en het maken van berekeningen, maar het is ook gevoelig voor menselijke fouten, zoals typefouten, formule fouten of verkeerd gekoppelde cellen.

Hier zijn enkele specifieke gevaren die kunnen optreden bij het gebruik van spreadsheets:

- **Menselijke fouten:** Aangezien Excel afhankelijk is van handmatige invoer, kunnen zelfs kleine fouten in gegevensinvoer of formules grote gevolgen hebben. Dit kan leiden tot onjuiste resultaten, verkeerde beslissingen en financiële verliezen. Het is essentieel om nauwkeurig te zijn bij het invoeren van gegevens en formules te controleren, om fouten te minimaliseren.
- **Verborgene rijen & kolommen:** Soms worden rijen of kolommen verborgen in Excel om gegevens overzichtelijker te maken. Het probleem is dat verborgen gegevens mogelijk niet worden meegenomen in berekeningen of grafieken, wat kan leiden tot vertekende resultaten en onnauwkeurige analyses.
- **Versiebeheer:** Bij het werken met Excel-bestanden in teamverband kan het moeilijk zijn om het overzicht te behouden over verschillende versies van een bestand. Wanneer meerdere mensen tegelijkertijd aan een bestand werken en wijzigingen aanbrengen, kunnen conflicten ontstaan, waardoor gegevens verloren gaan of verkeerd worden samengevoegd.
- **Gebrek aan gegevensvalidatie:** Excel biedt beperkte mogelijkheden voor gegevensvalidatie. Het ontbreken van strikte validatieregels kan leiden tot inconsistenties, onjuiste gegevensinvoer en problemen bij het analyseren van gegevens.
- **Beveiligingsrisico's:** Excel-bestanden kunnen gevoelige bedrijfsgegevens bevatten. Als er onvoldoende beveiligingsmaatregelen worden genomen, zoals het gebruik van zwakke wachtwoorden, het delen van bestanden via onbeveiligde kanalen of het ontbreken van toegangscontroles, kan dit leiden tot ongeautoriseerde toegang tot en misbruik van gegevens.



# Wanneer is het ontwikkelen van toepassingen op basis van 'code' een goede keuze?

Het is moeilijk om een algemeen antwoord te geven op wanneer het beter is om met code software te schrijven dan met no-code oplossingen, omdat dit afhangt van verschillende factoren, zoals de complexiteit van de toepassing, de vereiste functionaliteit, de schaalbaarheid en de technische expertise van het team. Er zijn echter enkele situaties waarin het gebruik van code meer geschikt kan zijn:

- **Weinig onderhevig aan veranderende omstandigheden:** Wanneer een toepassing eenmalig ontwikkeld dient te worden, en nadien wellicht weinig zal uitgebreid of aangepast moeten worden, is code geen slechte optie. Het grote voordeel van no-code platformen zit o.a. in de ROI die men kan bereiken door snel te kunnen ontwikkelen, en nadien snel te kunnen aanpassen.
- **Complexiteit:** Als je te maken hebt met een complexe en uitgebreide applicatie met specifieke vereisten, kan het gebruik van code meer flexibiliteit bieden. Sommige no-code platformen lossen dit op door de mogelijkheid te bieden om 'uit te breken' (escape hatch).
- **Schaalbaarheid:** Als je verwacht dat je applicatie in de toekomst aanzienlijk zal groeien en complexer zal worden, kan het gebruik van code een betere keuze zijn. Code geeft je de vrijheid om de architectuur en infrastructuur van de applicatie aan te passen en te schalen naarmate de behoeften evolueren. De vraag is dan hoe je die behoefte kan definiëren/valideren, alsook de afweging of de ontzorging van het no-code platform opweegt tegen het zelf beheren van de infrastructuur. Sommige no-code platformen kunnen in een private cloud ingericht worden, of op eigen 'servers' geïnstalleerd worden, om dit probleem te adresseren.
- **Technische expertise:** Als je team over de nodige technische expertise beschikt, kan het gebruik van code een voordeel zijn. Dit dient afgezet te worden tegen de backlog van het team.

Aan de andere kant bieden no-code oplossingen voordelen in termen van snelheid, gebruiksgemak en toegankelijkheid voor niet-technische gebruikers. Ze stellen mensen zonder programmeer achtergrond in staat om snel en gemakkelijk functionele applicaties en software te bouwen. No-code oplossingen zijn vooral handig voor applicaties, prototypes, interne tools en snel evoluerende projecten waar flexibiliteit belangrijk is.

In het algemeen is het belangrijk om de vereisten, complexiteit en toekomstige schaalbaarheid van je project zorgvuldig te evalueren voordat je beslist of je met

code wil werken of een no-code oplossing wil gebruiken. Soms kan een combinatie van beide benaderingen ook waardevol zijn, waarbij code wordt gebruikt voor specifieke aanpassingen en no-code tools voor snelle ontwikkeling van basisfunctionaliteit.

# Conclusie

Bij digitalisering is het de overweging of er een snelle tactische keuze gemaakt moet worden, dan wel, of er nood is aan een architectuur van software oplossingen om het bedrijf optimaal te ondersteunen. Laat je niet leiden door de prijs of het sexy imago van een tool: "a fool with a tool, is still a fool".

Bij een tactische keuze is de tool-selectie sterk gerelateerd aan de use case. Gaat het om een oplossing die een grote waarde (bv. commercieel voordeel) of groot risico (bv. privacy of reputatieschade) vertegenwoordigt, dan is de keuze voor professionele tooling evident. Betreft het een oplossing voor één enkele gebruiker, met weinig impact voor het bedrijf? Dan blijft een huis, tuin - en keukenoplossing verdedigbaar.

Wanneer digitale transformatie een thematiek is, dan dringt een strategische keuze zich op. Dan start de aanpak bij het begrijpen van de bedrijfsprocessen. Begin met het opdelen van de processen in 'core' en 'onderscheidende' processen om daarna de keuze te maken voor technologie. Overweeg hierbij om je manier van werken aan te passen aan de softwareoplossing die 80% van je noden afdekt. Vermijd grote aanpassingen van je standaard softwarepakket. Voor 'onderscheidende' processen is de kans groter dat maatwerk nodig zal zijn.

No-code platformen excelleren in snelheid van ontwikkeling en eenvoud van onderhoud van software toepassingen. Bovendien zijn ze meestal gehost in de cloud (ontzorging), zijn de applicaties bruikbaar op elk toestel (PC, smartphone, tablet) en zijn ze gebouwd om te kunnen integreren met andere software. Naast deze kenmerken zijn no-code platformen bijzonder nuttige instrumenten als brugfunctie tussen business gebruikers (bouwen van een MVP) en de IT afdeling (bewaken security, governance van data, performance, etc).