

This is how you do it...

Manage



Create



Launch



Low-code

En guide till tekniken som ökar hastigheten för applikationsutveckling dramatiskt och samtidigt möjliggör en harmonisering av applikationsportföljen.

 flowfactory

 IDG
CONNECT

Innehåll

Sammanfattning	3	Så fungerar low-code	9
Applikationsutveckling är inte snutet ur näven	4	Minska komplexiteten	10
Utvecklingen går åt rätt håll	5	Low-Code är inte No-Code	12
Olika typer av förenklad utveckling	6	Low-Code och digitaliseringsprocessen	12
Frågan är vad, inte hur	7	Framtiden är enklare	14
Skissa, bygg, klart!	8	Om FlowFactory	15



Sammanfattning

Applikationsutveckling har aldrig varit enkelt. Trots många, till viss del framgångsrika, försök att förenkla applikationsutvecklingen genom tiderna, är det fortfarande förknippat med höga kompetenskrav, höga kostnader och långa utvecklingscykler.

Behovet av nya applikationer bland företag och organisationer är samtidigt ständigt ökande. De behövs för att fortsätta vara konkurrenskraftig, för att driva utveckling inom digitalisering och för att skapa nya affärsmöjligheter.

Trots de behov som finns så är tillgången till kompetenta utvecklare betydligt mindre än efterfrågan på nya applikationer, vilket är en av anledningarna till långa leveranstider, höga kostnader och potentiellt missade affärsmöjligheter.

Low-code är en modell som syftar till att korta utvecklingstiden, sänka kompetenströskeln och inkludera fler människor i utvecklingen. Low-code bygger på grafiska utvecklingsverktyg och medger applikationsutveckling med ett minimum eller ingen programmering i traditionell bemärkelse. Samtidigt så är low-code väl lämpat för utveckling av avancerade och kraftfulla applikationer med möjlighet till integrationer med befintliga system, både inom organisationen och externa datakällor.

Low-code har, utöver möjligheten att korta utvecklingstiden och inkludera fler människor i utvecklingsarbetet, stor potential att minska komplexiteten inom organisationens applikationsflora och kodbas.

Ju mer en organisation nyttjar kraften i low-code, desto fler applikationer har möjlighet att köras på en gemensam plattform, med effekten av minskad komplexitet. Detta kan innebära stora förenklingar och minskade kostnader för organisationens IT-verksamhet.

Low-code kommer i olika typer och paketeringar. Gemensamt är dock möjligheten att öka takten och minska kostnaderna för organisationens applikationsutveckling, med ökad konkurrensförmåga, ökat tempo i digitaliseringen och minskad IT-komplexitet.



Applikationsutveckling är inte snutet ur näven



På marknaden för applikationsutveckling råder en kraftig obalans mellan behovet av nya applikationer, digitaliseringsfunktioner och digitala verktyg i allmänhet, och tillgången till folk som äger kunskapen att producera den kod som applikationerna i grunden består av. För att inte tala om bristen på folk som jobbar på design- och arkitektnivå. Den viktigaste faktorn i denna obalans är att programmering är svårt. Omvittnat skickliga arkitekter och utvecklare är det tjugoförsta århundradets hjärnkirurger och de vet sitt värde.

Samtidigt behövs det ju inte hjärnkirurger för att sy igen ett skärsår eller gipsa ett benbrott. Faktum är att de mesta bröd- och smör-utveckling är, om inte trivial, så åtminstone ganska rakt på sak ur en intellektuell synvinkel.

Men även denna del av utvecklingsprocessen är omgärdad av för höga trösklar för att "vem som helst" ska kunna bidra med kod. Även nyexaminerade script-kiddies från närmaste yrkeshögskola, ett slags cyber-proletariat på grundnivå, har trots allt ägnat ett par år i skolbänken och som marknaden ser ut idag är de inte gratis de heller.

Produktionskapaciteten för applikationer är alltså begränsad, på samma gång som behovet av nya applikationer ständigt ökar. Detta driver kostnader, uppmuntrar till nödlösningar och hämmar utvecklingen för företag och organisationer.

För företag och organisationer innebär den rådande situationen att utvecklingen hämmas och i många fall får negativa resultat.

- Dyrt. Med tanke på rådande situation förvånar det ingen att applikationsutveckling är förknippat med höga kostnader. Det finns gott om skräckexempel på skenande kostnader, inte minst inom offentlig sektor.
- Långsamt. Verksamheter är oftast i händerna på dem som har kunskapen, och om utvecklarna är fullbokade är det bara att ställa sig i kö. Kön är ofta lång. Time to market blir lidande.
- Haltande digitalisering. Digitalisering handlar i mångt och mycket om applikationsutveckling. Applikationer effektiviserar processer och flöden och integrerar verksamhetsområden och silos.

- IT mot resten. Pressen på IT-funktionerna är hård och på ökande, och till råga på allt ska IT spara pengar som alla andra. Risken är stor att IT uppfattas som bromsklossar när de måste prioritera bland utvecklingsprojekten. På beställarsidan stiger frustrationen.
- Shadow IT. Verksamheten orkar inte vänta på att IT ska förse dem med applikationer och verktyg, så man rundar IT och förser sig med det man behöver "på stan".
- Säkerhet. Gamla applikationer lever kanske inte upp till nya säkerhetskrav och Shadow IT är definitivt inte bra. Egentligen. Men man håller tyst.

IT-avdelningarna, med CIO i spetsen, är i en minst sagt besvärlig situation, ofta med motsägelsefulla krav. IT måste pressa tiden för time-to-delivery samtidigt som man är ålagda att "minska komplexiteten" för att kapa kostnaderna för driften av applikationer, hantera kodmassan effektivt och minska attackytorna av säkerhetsskäl, ibland kombinerat med allmänna nedskärningar av budget. Det går helt enkelt inte ihop.

Utvecklingen går åt rätt håll

En lösning som allt fler börjat titta på är den trend marknadsanalytikerna Forrester Research kallar Low Code och Gartner kallar High Productivity Application Platform as a Service (hpaPaaS). Ett mer generellt begrepp är Low Code Development Platform.

Low-code är en del av en utveckling vars drivkraft är att underlätta och snabba upp applikationsutveckling och programmering. De senaste årens teknikutveckling

har lett till att många ser att tiden och kostnaderna för applikationsutveckling kan kapas radikalt. Vissa talar om en revolution inom applikationsutveckling.

Ambitionen att korta utvecklingstiderna har genom åren lyckats till en viss del, men långt ifrån helt. Grafiska utvecklingsverktyg, designverktyg, digitala mindmaps och flödesscheman, enklare språk, WYSIWYG, Rapid Development, med mera, har alla varit, och är, faktorer som snabbat upp programmeringen signifikant.

Även målinriktade försök att skapa verktyg för applikationsutveckling genom enkla grafiska gränssnitt, inte minst för barn och skolor, har sett dagens ljus och den utvecklingen fortsätter.

Utvecklingen har inte gått obemärkt förbi stora aktörer som Google, Oracle, Salesforce, Microsoft med flera. Inom sfären för företagens produkter och tjänster finns möjligheter för kunder att utveckla applikationer skräddarsydda för kundens specifika behov, och det som utmärker dessa möjligheter ur ett utvecklingsperspektiv är att ingen, eller minimalt, med kodning behövs. Mycket har även gjorts på mobilappsidan för att förenkla utvecklingen på Android- och iOS-plattformarna.

Olika typer av förenklad utveckling

Idag finns det flera olika vägar att utvärdera när behovet av att ta fram en ny applikation eller förnya en gammal uppstår. Vägvalet beror dels av vilken typ av applikation det handlar om, hur unika behov och krav man har, hur detta behov matchar tillgängliga standardsystem på marknaden, men också på vilka krav och förutsättningar som finns gällande anpassning, tid och budget. I stort kan vi urskilja tre alternativa spår:

- Konfigurering av standardsystem (off the shelf). Inom ramarna för diverse administrativa system kan användarna skräddarsy applikationens funktionalitet. I många fall, speciellt för stora företag och organisationer med komplexa miljöer, är en "färdig" applikation sällan just färdig, utan kräver omfattande anpassningar, inte sällan tillägg av anpassad kod.
- Traditionell utveckling. Utvecklaren skriver kod i ett kodutvecklingsverktyg och kompilerar koden till färdiga programfiler. Det finns många hjälpmedel och förenklande metoder, men till sist handlar det om manuell produktion av kod.
- Low-code. Utvecklaren bygger applikationer, ofta för en specifik applikationsplattform, med hjälp av grafiska utvecklingsverktyg och ett minimum av kod. Ett synsätt är att utvecklaren "modellerar" applikationen från början, och lägger till den kod som inte finns i standardbiblioteket.

Begreppet low-code är brett hållet och rymmer många olika typer av lösningar och spelare. På senare år har en ny kategori low-code vuxit fram, så kallad "General Purpose". Dessa plattformar är breda i sin karaktär och kan hantera en mängd olika typer av applikationer. Low-code-aktörerna kan delas in i följande övergripande kategorier:

Nischspelare

Nischad low-code innebär att utgångspunkten är ett specifikt tillämpningsområde och passar bäst för applikationer inom en viss kategori eller användningsområde. Nischade low-code-plattformar kan exempelvis vara inriktade mot BPM (Business Process Management), ärendehantering eller specifika verktyg för mobila tillämpningar.

Ekosystem

Den här typen av low-code ingår i ett ekosystem av produkter och tjänster. Exempel är Salesforce's force.com och Google's App Maker. Här sker Low-code-utveckling inom ramarna för leverantörens ekosystem.

General Purpose

Dessa plattformar har ingen speciell inriktning på användningsområde eller typ av funktionalitet, utan används för att bygga generella affärsapplikationer. Inom kategorin general purpose har Forrester börjat tillämpa en ny kategorisering av plattformar; low-code för "proffs"-utvecklare och low-code för "affärs"-utvecklare. Lowcode för proffs är kraftfulla verktyg som kan användas för ett brett spektrum av stora och små tillämpningar, medan low-code för affärsutvecklare syftar till att ge personer utan programmeringskunskap möjlighet att själva konfigurera lite enklare applikationer.

När ska du välja att bygga applikationen med low-code?



Frågan är vad, inte hur

Är Low-code riktig programmering? Svaret beror naturligtvis på vad som läggs i uttrycket "riktig programmering". Svaret är nej om man menar manuellt skrivande av kod som kompileras till kompletta exekverbara applikationer, men svaret är ja om man menar att slutresultatet är applikationer som fullt ut fyller sin funktion för slutanvändaren. För slutanvändaren spelar det sällan någon roll hur applikationen kommit till, bara den gör det den ska på ett effektivt sätt.

Det kanske viktigaste att hålla isär när man ser till nyttan av low-code är att low-codeutveckling baseras på ett annat tänk. Klassisk programmering kräver en hel del fokus på hur du med hjälp av kod ska kunna lösa problem eller en uppgift?

Med low-code-utveckling behöver man inte tänka på hur något ska lösas tekniskt under ytan, utan kan helt fokusera på vad som ska lösas, alltså vad applikationen ska åstadkomma?

Att svara på frågan vad, och inte hur, öppnar i sig dörren för nya grupper av människor som tidigare inte kunnat bidra till applikationsutveckling. Det är betydligt enklare att svara på frågan vad man vill att en applikation ska göra, än att veta hur önskemålen ska implementeras efter att en design tagits fram. Low-code skapar både design och implementation med automatik, i samma process.

En stor styrka med low-code-utveckling är att många fler kan bidra till utvecklingen av applikationer. Genom grafiska gränssnitt och dra-och-släpp-funktionalitet sänks tröskeln väsentligt för dem som inte kan programmera. Som tekniken ser ut idag vore det

en överdrift att påstå att alla plötsligt kan utveckla applikationer; en förståelse för hur applikationer fungerar och data hänger ihop krävs trots allt, men för den stora majoriteten av arbetsinsatsen krävs ingen programmering. Detta öppnar för att beställaren av applikationen kan jobba närmare utföraren.

Skissa, bygg, klart!



Ett dilemma vid de flesta större utvecklingsprojekt är kommunikationen mellan beställare och utförare, eller snarare bristen på kommunikation. Risken för missförstånd är överhängande när beställaren, som inte är programmerare, försöker förklara applikationen och dess egenskaper för utvecklaren, och utvecklaren i sin tur försöker att översätta önskemålen till något som kan realiseras i kod.

Resultatet blir ofta att saker måste justeras i efterhand och projektet drar ut på tiden. Det kan bli många möten, med ett otal whiteboard-skisser, innan bägge parter känner sig nöjda.

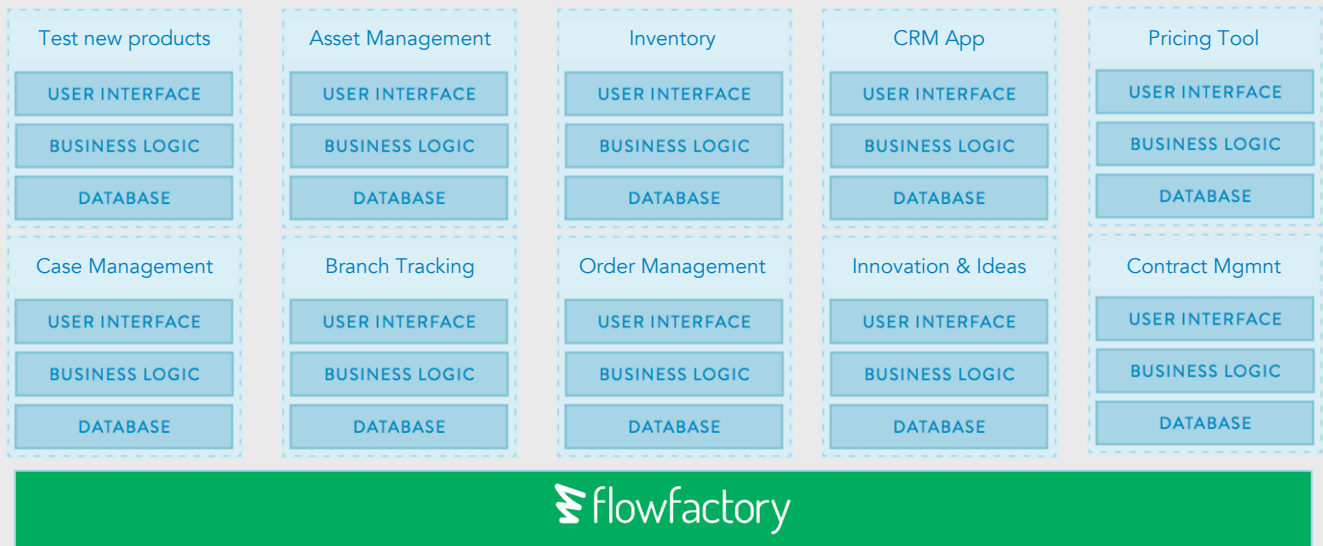
Detta ställer stora krav på beställaren, ofta så stora att beställare av applikationer är ett yrke.

Med low-code som verktyg framträder ett annat, och mer effektivt, scenario. Beställaren och utvecklaren kan mötas och en skiss av applikationen kan snabbt omsättas till något visuellt i ett digitalt verktyg. Det revolutionerande med low-code är att applikationen bokstavligen skapas parallellt med skissen.

Det är fullt möjligt att provköra många funktioner redan på detta stadium. Bägge parter ser resultatet direkt, vilket är enormt kraftfullt och sparar mycket tid.

Det är framförallt inom utvecklingen av kraftfulla affärsapplikationer som General Purpose low-code kommer till sin rätt. Vanliga exempel på sådana här applikationer är olika typer av säljsupporterande system, ärendehanteringssystem, produktionssystem samt administrativa system eller nya former av interna applikationer för att testa nya produkter och idéer.

Ett vanligt problem med de mer generella applikationerna, som finns att beställa från hyllan, är att de är byggda för att passa så många som möjligt och därmed kräver ordentligt med anpassning för att i slutändan passa kundens behov.



Ett vanligt problem med de mer generella applikationerna, som finns att beställa från hyllan, är att de är byggda för att passa så många som möjligt och därmed kräver ordentligt med anpassning för att i slutändan passa kundens behov.

“Onboarding” av en ny applikation marknadsförs ofta som snabbt, enkelt och billigt, men alla som varit med vet att det är när den första versionen är uppe som arbetet med att få applikationen att fungera och skapa värde egentligen börjar.

Det är ofta tilläggfunktionerna och specialanpassningarna som gör att kostnader och tid drar iväg ordentligt. Low-code gör det snabbt och enkelt att anpassa applikationerna för organisationens behov vid beställningen och vartefter de utvecklas, utifrån beställarens unika processer och typer av data. Här kan verkliga besparingar göras.

Så fungerar low-code

I praktiken är low-code ett verktyg där applikationer byggs i en visuell miljö med “dra och släpp”. Inom ramarna för plattformen hanteras både gränssnitt, logik och data vilket medför att allt som skapas och ändras publiceras som en riktig applikation i realtid. Själva logiken i applikationen byggs upp i form av flödesdiagram där varje steg i diagrammet representerar en aktivitet och övergångarna styr applikationens flöde. Denna typ av utveckling brukar kallas modellering snarare än programmering.

Low-codeplattformar rymmer stora bibliotek av standardfunktioner som används som byggstenar för att snabbt kunna få upp en skräddarsydd applikation. Om de tillgängliga komponenterna inte räcker till finns möjlighet att koda egna komponenter.

Till skillnad från enkla utvecklingsverktyg, som genererar en statisk applikation med hjälp av ett designverktyg, skapar low-code-plattformen snarare en definition av en applikation som tolkas och exekveras när den körs. I praktiken betyder det att man slipper kod som ska tas omhand och hållas uppdaterad. En stor fördel med att alla applikationer på plattformen kör i samma runtime är att plattformsuppdateringar slår igenom automatiskt för alla applikationer.

Utöver verktyget för att bygga applikationer innehåller low-code-plattformen också stöd för applikationsprojektets resa från idé till distribution av färdig applikation. Som standard levereras en low-code plattform som PaaS- lösning (Platform as a Service) via molnet men kan även installeras lokalt i kundens egna driftsmiljö (on premise). Med en molnlösning blir all teknik omhändertagen och applikationerna fullt skalbara.

Många applikationer börjar sitt liv på en whiteboard; det revolutionerande med lowcode är att om vi byter ut tavlan mot ett grafiskt low-code-verktyg och ritar strukturen för applikationen i verktyget i stället, har vi inte bara tagit fram en design, vi har faktiskt redan skapat större delen av applikationen.

Fördelen med detta sätt att utveckla applikationer är tämligen uppenbara. Dels ger det grafiska tillvägagångssättet en betydligt bättre överblick och förståelse för hur de olika delarna av en applikation hänger ihop, dels tar det avsevärt mindre tid. Faktum är att när utvecklaren väl har vanan inne går det tre till fem gånger så fort att producera applikationen med low-code än med traditionell utveckling. Och inte minst, givet att utvecklingsverktyget i sig inte är alltför komplicerat för den oinvidge, öppnas dörren för många fler att utveckla applikationer, eller i vart fall delta mycket närmare i processen än tidigare.

Projektportal

Hantera projekt



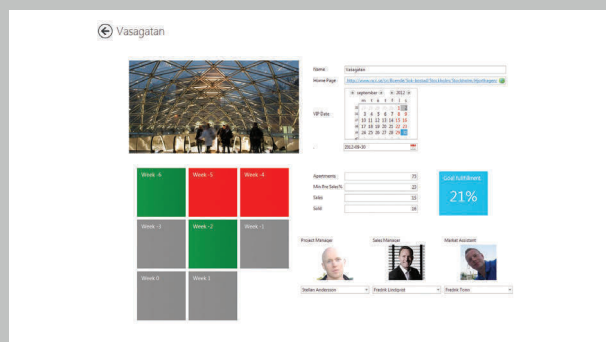
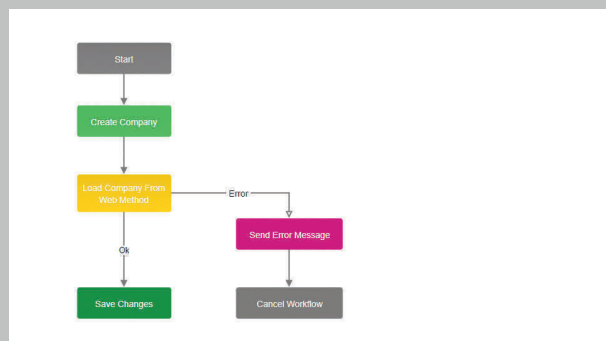
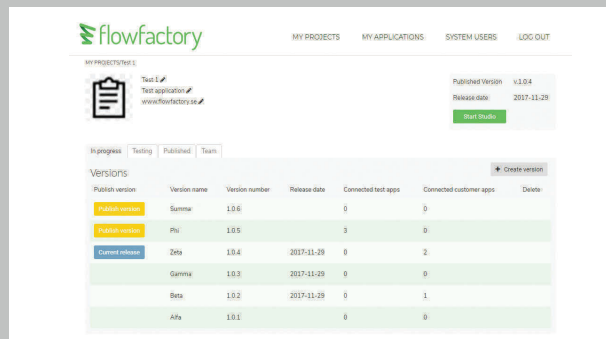
Studio

Skapa och förfinas applikation



Applikation

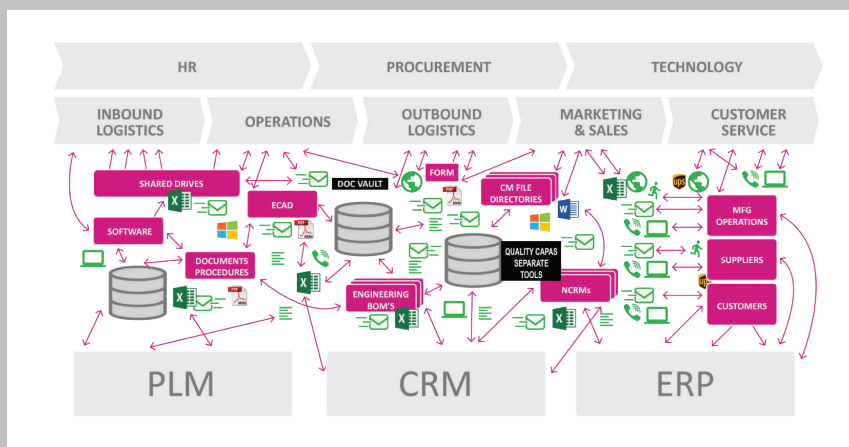
Leverera affärsnyttan



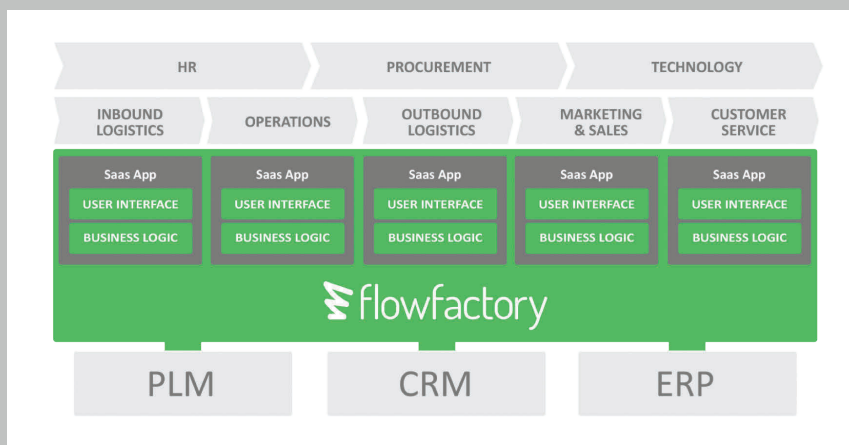
Minska komplexiteten

En annan styrka med low-code är att plattformen kan ligga som ett lager ovanpå organisationens befintliga system med standardiserade integrationer. På denna plattform kan organisationen sedan snabbt och enkelt utveckla skräddarsydda applikationer för de olika funktionerna och rollerna inom organisationen.

Med hjälp av en modern low-codeplattform så kan organisationen skapa en arkitektur baserad på mikrotjänster istället för monolitiska lösningar med högre återanvändbarhet och skalbarhet. För organisationens IT minskas då komplexiteten avsevärt genom att de slipper hantera mängder av kod, plattformar, tekniker, integrationer och kostsamma underhåll, och med tiden i stället kan samla många av sina applikationer på samma plattform, där all nödvändig teknik är omhändertagen. Uppdateringar av plattformen slår igenom generellt.



Generisk nulägesbeskrivning: Bilden visar en vanlig situation i många organisationer; hundratals olika, små som stora applikationer och stödsystem driver mot ökad komplexitet och kostnad.



Konceptbild: En lowcode plattform gör det möjligt att påbörja resan mot en renare och mer snabbfotad applikationsportfölj. "Feta" integrationer och affärslogik återanvänds i olika applikationer vilket minskar komplexiteten över tid, allteftersom man växlar in uttjänta applikationer och bygger nya på enoch samma plattform.



Low code-plattformen trollar inte bort befintliga applikationer eller befintlig kodbas, men komplexiteten minskar över tid ju fler applikationer som skapas på plattformen, i synnerhet om de nya applikationerna ersätter gamla. Däremot ökar inte komplexiteten hur många plattformsapplikationer som än skapas - det är en och samma plattform vare sig den härbärgerar en eller ett hundra applikationer.

För IT-avdelningen är detta högst välkommet eftersom de annars måste räkna med ökad komplexitet, vilket i sig skapar visst motstånd mot nya applikationer i organisationens redan välfyllda flora av kod, plattformar, licenser och så vidare. Detta gäller oavsett om en applikation köps in "från hyllan", utvecklas inom organisationen eller utvecklas av externa konsulter. Gamla applikationer som kanske bara ett fåtal personer använder, men som kan vara ack så viktiga för dem, kan i sinom tid ersättas av low-code-utvecklade applikationer utan den ökade komplexitet man annars skulle förvänta sig.

Low-Code är inte No-Code

Enkla applikationer kan helt och hållet byggas inom ramarna för low-code, men man kommer till syvende och sist inte runt traditionell programmering, om så bara till viss del, när det kommer till mer kraftfulla applikationer som måste integreras med andra applikationer, eller exempelvis nyttja API:er mot externa resurser.

Det är inte heller speciellt märkligt med tanke på artrikedomen bland de system man vill att ens applikation ska integrera sig med, och det gäller ju även traditionell utveckling. Integrationerna skapas genom att en utvecklare kodar de bibliotekfunktioner som behövs. Dessa funktioner läggs till i biblioteket och finns sedan tillgängliga i det grafiska utvecklingsverktyget, och kan givetvis återanvändas framöver. Low-code är alltså inte no-code. No-codeutveckling begränsas till de komponenter som leverantören tagit fram, vilket i och för sig kan räcka för enklare applikationer och prototyper. På en low-codeplattform är det möjligt att lägga till egen kod, som då blir färdiga komponenter i utvecklingsverktyget. Detta ger företag med komplexa behov frihet att bygga stora system utan begränsningar.

Kostnaden för att utveckla i low-code, jämfört med att utveckla helt manuellt, blir med andra ord betydligt lägre. Utvecklare kan också spara en massa tid genom att det går väldigt fort att skapa enkla standardkomponenter som inmatningsfält och knappar, och i stället lägga fokus på integrationer och specialanpassningar, där den dyrbara utvecklarkompetensen kommer till sin fulla rätt.



Low-Code och digitaliseringsprocessen

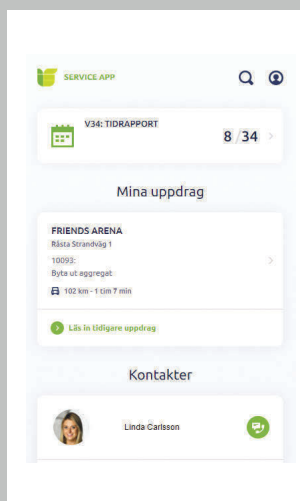
Low-code kan kraftigt höja produktiviteten för IT-avdelningar och skapa manöverutrymme för IT att leda hela organisationens digitaliseringsprocess. Digitalisering är att automatisera och rationalisera processer och flöden med digitala system och verktyg, och det ligger i low-codeteknikens DNA att stödja denna typ av utveckling. Som vi skrev ovan; om man byter en whiteboard mot ett utvecklingsverktyg för low-code, ritar i princip samma sak, så är utvecklingsprocessen för den tänkta applikationen redan igång.

Det är när IT-chefen ser hur efterfrågan på applikationer från verksamheten ökar och det känns omöjligt att kunna svara upp mot denna, som IT-chefen ska tänka low-code. Givet samma resurser kan IT-avdelningen, tillsammans med beställarna, utveckla applikationer 3-5 gånger fortare än med traditionell programmering.



The screenshot shows a mobile app interface for 'Offert Folding test'. It includes a table with columns for 'Position', 'Namn', 'Kundpris', 'Kundpris', and 'Pris tillsammans'. The table lists items like '1LA7080-4A412-Z' and '6ES7315-6FF04-0A80'.

Position	Namn	Kundpris	Kundpris	Pris tillsammans	
1		266,022.12		3,110.22	
1.001	1LA7080-4A412-Z 3-fas motor (IP55, 4-polig, sterisk 80 M)		1.00	80.00 % (manuell)	836.10
1.002	6ES7315-6FF04-0A80 CPU315F-2DP till 57-300, 384KByte, gråns		3.00	40.00 % (manuell)	1,635.00
2	Pos 2			16,453.07	
3	Pos 3				
4	Pos 4			244,458.83	



The 'Lägg till bild' (Add image) form includes a photo of a washing machine and a checklist for components. The checklist items are: Blandare, Fuktskada, Tvättmaskin, Handduktork, Vattenlås, Radiator, Golvbrunn, Fönster, Undertak, Värmegolv, and Ej godkänd inst. The 'Vattenlås' and 'Ej godkänd inst.' items are checked.



Framtiden är enklare

Low-code, speciellt General Purpose Low Code, spås en fortsatt stark framtid. Utvecklingen av tekniken har varit tämligen lågmäld fram till för ett par år sedan, men liksom AI tippar den samlade teknikutvecklingen över en tröskel och plötsligt går det undan. Low-code är uppmärksammas som en teknikutveckling att följa på nära håll av analytiker som IDC, Gartner och Forrester.

Teknikutvecklingen kring low-code förväntas löpa längs flera spår. Dels kommer utvecklingsverktygen bli mer användarvänliga för verksamhetsansvariga, dels kommer miljöerna och plattformarna för runtime breddas och gå mot mer "General Purpose" än idag.

Till det bör läggas att "riktig" programmering i allt högre grad rör sig i riktning mot högre grad av automatisering och förenkling. Nya moderna språk för mobil och webb drar alla tveklöst mot förenkling; det är ofta hela syftet.

En kategori som kommer vinna stort på utvecklingen är startups. I princip måste alla nya projekt inom den digitala arenan involvera utvecklare. Så lär det även bli ett tag framöver, men utsikten att kunna rita sin affärsprocess i ett utvecklingsverktyg och därefter ha fått fram en färdig applikation kan göra low-code stort. Mycket stort. Low-code kan bli för folk med en idé, vad 3D-skrivare är för prototyp tillverkning.

Om FlowFactory

Flowfactory är en svensk low-code-plattform för kraftfulla affärsapplikationer – den första av sitt slag på den nordiska marknaden. Plattformen hanterar både gränssnitt, logik och data och levereras som en molntjänst eller för intern installation (on-premise). Flowfactory erbjuder också attraktiva "komma-igång" tjänster, med egna resurser eller via auktoriserade partners. Bolaget är i stark tillväxt och på plattformen finns idag kunder som, Siemens, Region Stockholm, Länsstyrelserna, LoginHasselberg, Resursen m fl.

hello@flowfactory.
www.flowfactory.com

