

Göra eller köpa kod

Välj det bästa av två världar



Inom mjukvaruområdet finns ett gap mellan komplexitet och produktivitet. Dessutom finns en växande underhållsbörda från existerande mjukvara. Det betyder att antalet mjukvaruutvecklare hela tiden måste öka för att bibehålla produktiviteten. Med 25 procent årligen säger en siffra. Lägg till denna börda att det finns en brist på begåvningar inom mjukvaruområdet.

Här är ett tips som kan minska pressen: använd en smart strategi för inköp av mjukvara.

Alla mjukvaruföretag behöver ta beslut på vilka delar av mjukvaran som ska utvecklas internt och vilka delar som ska anskaffas externt – göras eller köpas?

VI REKOMMENDERAR att sådana beslut baseras på frågeställningar av följande slag:

- Ligger det någon konkurrens fördel i att utveckla en viss mjukvarukomponent på egen hand? Kan vi kanske arbeta bättre, snabbare, med mindre resurser eller med högre kvalitet än konkurrenterna?
- Är komponenten en del av vår kärnkompetens?
- Är komponenten strategiskt relevant, det vill säga bidrar den till vår usp, vår unika säljproposition?
- Hjälper komponenten till att särskilja vår produkter och tjänster från konkurrenternas?

OM SYSTEMARKITEKTUREN är väldefinierad kan beslutet om att göra eller köpa fattas för varje komponent för sig. Summan av alla dessa göra-eller-köpabeslut är vad vi kallar för en "inköpsstrategi".

Eller vänta förresten – är det verkligen allt som finns att säga om inköp? Finns det faktiskt bara två alternativ?

Det tror inte jag. Här döljer sig en djupare frågeställning än bara göra-eller-köpa. Nämligen – hur? Låt oss titta närmare på några olika strategier som finns vad gäller att införskaffa mjukvara och vilka för- och nackdelar de har.

1. Intern utveckling ger exklusivitet

Första alternativet är att utveckla mjukvarukomponenten internt – hela vägen från kravställning och specifikation till genomförande, integration, test och underhåll.

Om man gör det får man fullständig ensamrätt till mjukvaran. Dessutom får man högsta möjliga sekretess. Och så kan den skraddarsys efter dina behov. Och så finns all relevant kompetens kring komponenten samlad internt på ditt företag.



Av Robin Schubert, Baselabs

Robin Schubert är vd på Baselabs i Chemnitz, Tyskland, som är specialiserad på sensorfusion. Robin Schubert blev kvar i sex år på Technische Universität Chemnitz efter att ha doktorerat i IT och elektroteknik. Han arbetade då som chef för en forskargrupp inom området datafusion.

Baselabs är baserat på hans forskning, och grundades år 2012 av honom och tre partners.

Utmaningen är att det kräver en personalstyrka av kvalificerade programmerare. Samt tid – en ännu knappare resurs. Tid inte bara för utvecklingsfasen utan även för underhåll och support efteråt.

Den som bestämmer sig för att utveckla sin mjukvara på egen hand får på köpet även i knäet att underhålla denna programvara i framtiden, så länge som den används.

Man underskattar ofta inte bara hur mycket resurser som krävs för utveckling utan även hur mycket resurser som krävs för un-

törshantering, systemintegrering och systemtest fortfarande måste hanteras internt.

Ur tids- och kostnadsperspektiv är den totala insatsen jämförbar med intern utveckling. Det finns möjligen en möjlighet att spara kostnader om konsulten finns i en låglöneregion, noter skillnaden mellan "offshoring" och "nearshoring".

Den viktigast fördelen med den här strategin är att avtal kan avslutas snabbt och enkelt om planer ändras. Det ger mer flexibilitet. För stora företag kan det faktiskt ibland vara lättare att kontraktera ett externt företag än att utnyttja interna resurser. Med ett externt kontrakt kan det vara lättare att genomdriva att leverantören uppfyller avtalet.

En nackdel är att du får mindre kontroll över utvecklingsprocessen och över utvecklarnas kvalitet. Att anlita konsult (särskilt offshore) kräver typiskt en grundlig process för kvalitets- och leverantörskontroll. Dessutom blir du alltmer beroende av konsulten och får därmed en allt sämre förhandlingsposition, allteftersom projektet fortskrider.

Den mest kritiska faktorn är dock underhåll. Eftersom mjukvaran inte utvecklats helt inom ditt eget företag blir den svår och dyr att underhålla på egen hand. Att få entreprenören att åtgärda ett problem efter leverans kan vara svårt, eftersom teamet som levererade programvaran kan ha ändrat utseende med tiden.

3. Licensierad mjukvara spar tid och pengar

Om mjukvaran redan finns på marknaden kan du köpa en licens (en rätt att använda den). Utvecklingstid och resursanvändning minskar då avsevärt eftersom komponenten ju redan finns och inte behöver utvecklas överhuvudtaget.

Den fundamentala fördelen med mjukvara är att du bara behöver utveckla ett enda exemplar. Det kan mångfaldigas i oändlighet och användas av godtyckligt många.

Licensiering som inköpsstrategi är ett sätt

"Det är värt besväret att granska sin inköpsstrategi"

derhåll. En IEEE-enkät gav statistiken att det genomsnittliga budgetöverdraget för mjukvaruprojekt låg på hela 89 procent.

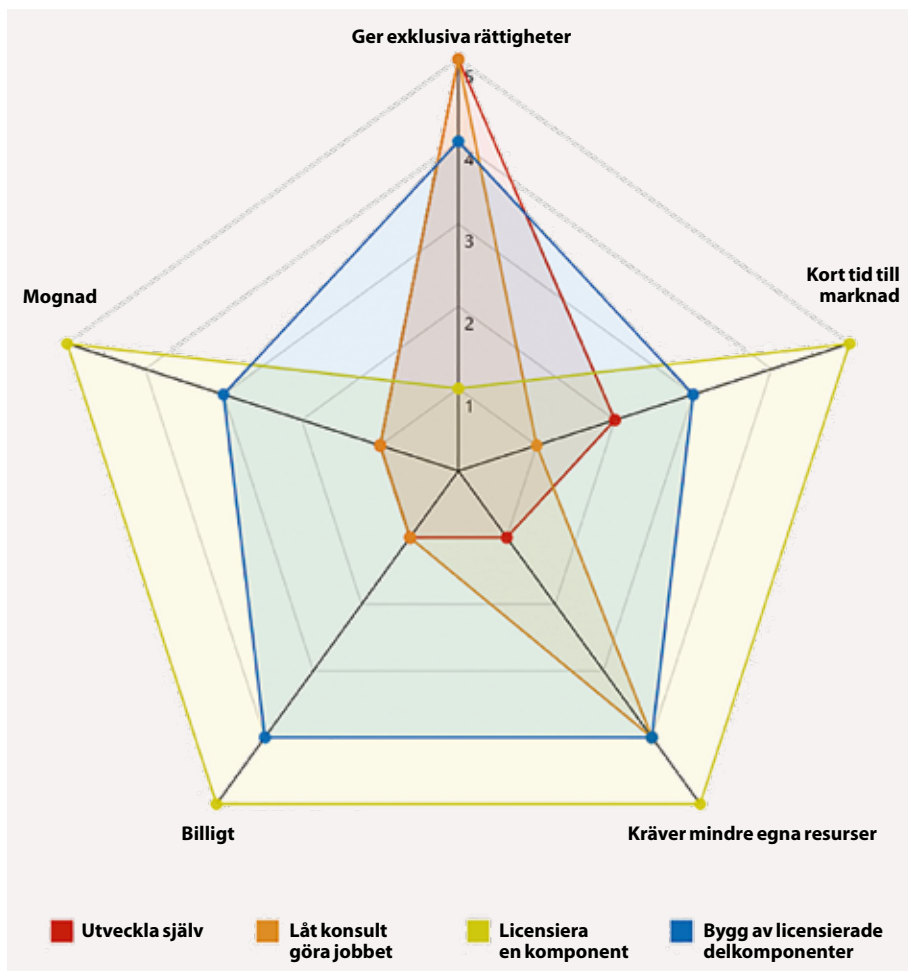
Internutveckling förknippas typiskt med oberoende. Det stämmer ur immaterialrättsligt perspektiv. Däremot uppstår ett annat beroende – personal. I de flesta team är expertisen inte jämnt fördelad. Risken finns alltid att en central utvecklare väljer att lämna projektet.

Dessutom begränsar internutveckling din frihet att utnyttja utvecklare effektivt, eftersom varje ny mjukvarukomponent kräver ytterligare utvecklare för underhåll.

2. Extern utveckling ger flexibilitet

Det näst vanligaste alternativet är att lägga utvecklingen på en extern konsult. Vad gäller exklusivitet och IP-rättigheter gäller detsamma som för internutveckling.

Fördelen är att det krävs färre interna utvecklare, även om kravutformning, leveran-



att dela upp utvecklingskostnaderna (eller kostnader plus leverantörmarginaler om man ska vara noga) mellan flers kunder, vilket ger en dramatisk kostnadsfördel jämfört med internutveckling och kontraktutveckling. Av samma skäl är även kostnaderna för underhåll och support vanligen drastiskt lägre.

En ytterligare fördel är att mjukvaran redan är testad av andra användare. Det sänker risken för buggar och (givet att testerna är väldokumenterade) minskar avsevärt dina egna testbehov. Den är bevisad-i-bruk som det heter inom de områden där funktions-säkerhet tillhör kravbilden.

Nackdelen är att du inte får monopol på de immateriella rättigheterna utan att dina konkurrenter kan använda samma komponenter. Å andra sidan – om mjukvaran finns på marknaden kan ju dina konkurrenter använda den oavsett vad du gör.

En möjlig begränsning med licensierad mjukvara är att vissa leverantörer inte erbjuder den "white-box", det vill säga med full tillgång till komplett källkod, vilket är något du behöver för att kunna bygga upp intern kompetens. Grundfrågan är dock om mjukvaran du söker finns på marknaden eller inte.

Vad vi lärt så långt

• Intern och extern utveckling är egentligen två sidor av samma mynt. I själva verket tycker jag inte att det är en "mjukvaru-

strategi" att lägga utveckling på en extern konsult, utan snarare en "utvecklarstrategi". Ska jag anställa utvecklare själv eller anlita ett mjukvaruföretag som anställer dem för min räkning?

• Med tanke på mjukvarans immateriella natur är den viktigaste faktorn för att skära ner kostnaderna att dela investeringen med andra genom att licensiera den. Den som utvecklar mjukvara internt eller via konsult får alltid högre kostnader än konkurrenten som köper en ickeexklusiv licens till en redan existerande komponent. Du accepterar ett prispåslag som du måste kunna motivera med att det finns en högre betalningsvilja hos kunden.

DET FINNS ETT FJÄRDE ALTERNATIV som enligt min mening ofta förbises utan ordentlig motivering. Det är att kombinera exklusiv kod (det strategiska valet) med licensiering (det kostnadseffektiva valet). Låt mig förklara!

4. Det bästa av två världar

Det är ganska vanlig att mjukvara med exakt rätt funktionalitet inte finns på marknaden. Många chefer drar då slutsatsen att komponenten ifråga måste utvecklas internt. I det läget kan det vara värt att syna arkitekturen lite mer i sömmarna.

Hela komponenten kanske inte finns att köpa. Däremot kan du ofta pussla samman

den med hjälp av delkomponenter, rutiner eller funktioner som finns kommersiellt. Mjukvaran i sig kan vara strategiskt relevant utan att delarna är det. Ofta ligger konkurrensfördelen i att intelligent kombinera befintlig teknik snarare än att skapa en komplett innovation från ett blankt ark.

Komersiellt tillgängliga bibliotek och utvecklingsmiljöer (SDK:er) kan användas för att utveckla konkurrenskraftig mjukvara. På det sättet kombinerar du intern utveckling med licensiering.

Det finns sådana bibliotek inom flera områden och de används i stor utsträckning, som TensorFlow för djup maskininlärning, OpenCV för klassisk bildbehandling – och Baselabs Create Embedded för säker sensorfusion i inbyggda system.

ATT ANVÄNDA DELKOMponenter ger samma fördelar som licensieringsstrategin. De finns att hämta direkt, de kostar en bråkdel, de är testade och de splittar notan för underhåll på många användare.

Funktionalitet på denna lägre, generiska nivå har ofta betydligt bredare tillämpning vilket betyder att antalet användare som delar på notan för utveckling och underhåll är fler än för en specialiserad komponent.

Genom att kombinera funktioner från bibliotek eller en SDK kan du utveckla mjukvarukomponenter som du har exklusiva rättigheter och exklusiv tillgång till. Det innebär dessutom att relevant kompetens hamnar i ditt företag och att allt kan skräddarsys efter egna behov.

Priset för strategin är extraarbetet med att identifiera lämpliga bibliotek och SDK:er, och att verifiera att de uppfyller kravbilden. Du behöver en mer detaljerad inköpsstrategi och mer teknisk expertis i inköpsrollerna. Å andra sidan kan sådana satsningar vara ett effektivt sätt att utnyttja knappa interna utvecklingsresurser.

LÅT OSS GÅ TILLBAKA till den översiktliga bild vi presenterade tidigare. Genom att jämföra alternativen kan vi visualisera deras respektive för- och nackdelar:

Komponentlicensiering är klart att föredra om den möjligheten existerar, samtidigt som egen utveckling kan användas som komplement. Bibliotekslicensiering kombinerar fördelarna med båda världarna.

Slutsats: Det är värt besväret att granska sin inköpsstrategi

Hur din strategi utfaller beror på vilken detaljnivå den tillämpas på. En analys på hög nivå ger begränsade alternativ och är ofta otillfredsställande.

Genom att lägga tid på en mer detaljerad analys når du upp till en helt ny strategisk nivå som kan ge klara konkurrensfördelar. Samtidigt minskar du kostnad, tid-till-marknad och behovet av interna utvecklare. Upphandling är en kraftfull hävstång – det är upp till dig att välja hur du vill använda den. ■