

RIV PETER ANTMAN

PYRA
MID-
ERNÄ
IGEN

MED HJÄLP AV AGILE OCH LEAN

Riv pyramiderna igen

med hjälp av Agile och Lean

Peter Antman

Denna bok är till salu på <http://leanpub.com/pyramiderna>

Denna version publicerades den 2014-07-24



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2013 - 2014 Peter Antman

Twittra denna bok!

Hjälp Peter Antman genom att berätta om den här boken på [Twitter!](#)

Den föreslagna hashtaggen för den här boken är [#pyramiderna](#).

Ta reda på vad andra säger om den här boken genom att följa den här länken till en sökning på Twitter:

<https://twitter.com/search?q=#pyramiderna>

Innehåll

Om författaren	1
Andra medverkande	2
Revolution i skymundan - det nya arbetslivet	3

Om författaren

Peter Antman är programmerare och författare. Han har spenderat större delen av det senaste decenniet till att bygga system som bygger system, det vill säga växa organisationer som är verkligen bra på att skapa mjukvara, i roller som utvecklare, arkitekt, CTO och utvecklingschef.



Peter Antman

Idag arbetar han som agil management-coach i konsultbolaget [Crisp](http://crisp.se)¹. En gång i tiden arbetat han mest som journalist och förberedde en forskningskarriär inom sociologi och idéhistoria, men upptäckten av Linux och fri mjukvara fick honom att byta bana. Han bloggar han aktivt för [Crisp Blog](http://blog.crisp.se)² och Smartbear.

Peter har bland annat tidigare publicerat:

- *Vandringsturer i Padjelanta*, Vildmarksbiblioteket (2012)

¹<http://crisp.se/peter.antman>

²<http://blog.crisp.se/author/peterantman>

- *Olof Palme - reformista sin fronteras* (med Pierre Schori), Cedecs
- *Barn och äldreomsorg i Tyskland och Sverige*, Socialdepartementet (1996)
- *Olof Palme - den gränslöse reformisten* (med Pierre Schori), Tiden (1996)
- *Den ohälsosamma vården?* (med Rolf Å. Gustafsson), Karolinska institutet (1995)

Andra medverkande

Omslaget är gjort av Lisa Zachrisson, Zachrisson grafisk form.

Revolution i skymundan - det nya arbetslivet

När mår vi som bäst, när uppnår vi bra resultat, när är vi som mest innovativa och kreativa på jobbet?

Lite i skymundan pågår något av en revolution av vårt sätt att arbeta i många av våra IT-företag, särskilt de som arbetar med mjukvaruutveckling. I centrum står en ny och ljusare syn på människan: vi är inte i första hand bångstyra egoister som måste styras och kontrolleras. I själva verket både mår och arbetar vi bäst när vi i mindre grupper tillsammans med andra själva tar ansvar för vårt arbete. Det är som om vi är designade för att arbeta tillsammans i grupp mot gemensamma mål.

Som så mycket annat livsförändrande de senaste decennierna är det till IT-branschen och programmerarna man i första hand får vända sig för att hitta framtiden. Den demokratisering av produktionsformerna i kunskaps ekonomin som drivits så starkt från programmerarkulturen på den amerikanska västkusten, internet, persondatorn, webben, öppen källkod, nedladdning av musik och film, söker sig nu in i företagen och vänder dem både ut och in och upp och ner.

På gräsrotskonferenser, i nätfora och i management-litteratur äger vår tids kanske mest avancerade och levande diskussion om hur man bäst organiserar arbete rum. Här kritiserar man Taylorism, talar för självstyrande grupper och fritt förfogande över arbetstiden eller propagerar för att det är dags att återinföra lönegrader.

Inom mjukvarubranschen har sedan mitten av 90-talet arbetssätt baserad på värderingar som betonar samarbete och dialog framför hierarkier och order vuxit sig stark. Med ett samlingsnamn brukar

vi tala om dem som agila metoder. Det som började bland engagerade programmerare och i mindre produkt- och konsultbolag, är idag det självklara sättet att jobba inom moderna företag som Facebook och Spotify och anammas idag också av gamla stela kolosser som telekomföretag och banker.

Så här motiverar bland annat managementkonsulten David Andersson - före detta chef på Microsoft - sin metodologi för uthålligt arbete: "I wanted ... to improve the conditions that were causing stress-related health issues in developers as young as in their 20s".

Det är förstås inte någon välgörande verksamhet som ligger bakom detta. Många företag i IT-branschen, inte minst Google, har närmast extrema vinstmarginaler. Trots ökad global konkurrens från låglöneländer råder det ännu brist på programmerare och arbetsgivare måste ofta stå på tå för att behålla sin arbetskraft.

Som alla vet är det också en bransch med ofta extrem förändringstakt och därmed krav på hög utbildning och rask vidareutbildning. Intressanta saker händer med maktstrukturer när de anställda är bättre utbildade än sina chefer.

Men det handlar inte enbart om makt och överflöd, utan också om själva resultatet av arbetet. För jag vågar nog påstå att den grundläggande drivkraften trots allt varit och är att förbättra just vad man åstadkommer: att öka kvaliteten och innovationsförmågan, minska slöseriet med resurser, och – med närmast en klyscha – ge kunden det den verkligen vill ha. Och en fundamental insikt i att det inte går att tvinga fram detta.

Historien om IT-projekt är nämligen till stora delar en historia om misslyckanden. Försenade leveranser, buggiga system och avsaknad av efterfrågad eller rentav konstig funktionalitet är legio. Enligt industrisiffror är möjligen så mycket som 60 procent av världens producerade mjukvara oanvänd(bar).

Idag råder närmast total konsensus i programmerarvärlden att alla dessa misslyckanden beror att man inte förstått vad systemutveck-

ling egentligen är. Den traditionella utgångspunkten har varit att några få smarta personer kan tänka ut hur ett system ska fungera och göra en ritning över det. Genom att bryta ner systemet i mindre delar kan man skapa funktionell arbetsdelning. Några få tänker och många utför. Ett underliggande antagande är att det inte uppstår någon relevant kunskap vid själva utförandet som kan påverka den ursprungliga planen.

Just så här, menade två inflytelserika japanska forskare på 80-talet, är de flesta västerländska företag organiserade: en liten elitklick som tänker åt alla andra som utför.

I kontrast till detta visar de att en rad japanska företag varit mycket lyckosamma i att bedriva produktutveckling just genom att inom företaget ta fram, bevara och sprida ny kunskap genom ett experimenterande lagarbete.

I början av 90-talet tog den före detta stridsflygaren och mjukvaru-entreprenören Jeff Sutherland över de japanska forskarnas ruggby-baserade metafor för hur framgångsrika produktföretag arbetar när han skapade arbetsmetodiken Scrum: självorganiserande team som arbetar mot ett gemensamt mål och "springer" mot målet genom att hela tiden kasta bollen fram och tillbaka. Världen över har mjukvaruföretag anammat detta sätt att arbeta och med det radikalt nya sätt att förhålla sig till styrning och belöningssystem.

Programmering och systemutveckling handlar nämligen väldigt mycket om lärande och skapande av ny kunskap. Modern kognitionsforskning bekräftar på många sätt hur viktigt informellt lärande är för kunskapsstillväxt, och gör ett viktigt tillägg: man kan inte tvinga fram den typen av lärande. I stället måste den snarare frigöras, eftersom den beror på känslorna, på vår inre lust eller motivation.

Dan Pink, en gång talskrivare till All Gore och uttolkare av dessa nya sätt att arbeta visar i en serie föreläsningar och den fascinerande boken *Drive* hur dåligt externa belöningar fungerar för att driva på människor, förutom i väldigt enkla och avgränsade situationer.

Vår förmågan till problemlösning minskar nämligen när externa belöningar introduceras. Man blir så fokuserat på moroten att man tappar i förmågan till problemlösning. Det är helt enkelt skadligt att använda externa belöningssystem för att påverka den inre motivationen! "This is", skriver Pink i Drive, "one of the most robust findings in social science – and also the most ignored".

Med andra ord. Hierarkiska företag som tror att kunskap kommer uppifrån och sprids ut i organisationen och som bygger på belöningssystem som med piska och morot påverkar den yttre motivation kommer helt enkelt inte lyckas lika bra som de som i stället vågar befria sina medarbetare. När, menar Pink, människor får kontroll över sina uppgifter, sin tid, sin teknik och sitt team så slutar det inte med att de slöar eller gör helt andra orelaterad uppgifter, i stället ökar lusten att arbeta, att uppfinna, att göra något bra.

Men, det är inga småsaker man måste ändra på för att organisera sin verksamhet efter dessa principer.

Ta autonomi. Den säger oss att det är bara om man har makt att själv avgöra med vad, när, hur och med vilka man arbetar som man uppfattar sig själv som autonom. Och det är bara om jag uppfattar mig själv som själv-styrande, som jag arbetar av inre drivkraft.

Eller ta skicklighet. Människan tycks må som bäst när hon arbetar precis vid gränsen för sin förmåga, inte utmattande men utmanande och lärande. Numera pratar man ofta om ett tillstånd av "flow". Men hur skapar man förutsättningarna för det utan att bränna ut människor?

Och hur skapar man en känsla av ändamål? För att frigöra energin i en grupp människor som har rätt och möjlighet att styra över sig själva behövs vettiga och rimliga mål. Men, som Pink skriver, "Wealth maximization lacks the power to fully mobilize human energies". Kan kommersiella företag verkligen skapa detta?

Mycket talar för det, åtminstone i begränsad utsträckning. Bland

programmerarna i IT-branschen, har det utvecklats en rad arbetsmetoder för att skapa både autonomi och arbetsflöden som gör det möjligt för folk att i små team arbeta i ett rimligt men utmanande tempo med stor variation, korta feedbackloopar och med inbyggt lärande, som hos Google där utvecklarna faktiskt fritt förfogar över 20 procent av sin arbetstid, som de kan ägna åt egna projekt.

2007 tillträdde jag som utvecklingschef för ett internationellt produktbolag inom mediabranschen. Mitt mål var att hjälpa utvecklingsavdelningen att bli en del av denna förändring. Under fem års tid ändrade vi i grunden vårt sätt att arbeta. I centrum av den förändringen stod självstyrande team och högt flyt i arbetet. Jag vågar påstå att vi blev världsklass på agile (se sidotext om det agila manifestet).

Om verkligheten är dynamisk och komplex (så som till exempel kundbehov ofta är) så finns den verkliga kunskapen om denna ute i tentaklerna hos de som arbetar nära "verkligheten". I den mån ett företag, en organisation eller en grupp vill dra nytta av denna kunskap på ett snabbt och effektivt sätt så måste möjligheten att fatta beslut utifrån kunskapen också vara spridd. För att kunna fatta beslut måste man kunna ta ansvar för det man gör och för att kunna ta ansvar behöver man ha makt. Ett av de mest grundläggande kraven för att ha makt är att äga sin egen tid, att kunna styra över sitt arbete.

Men för att nå dit räckte det inte med att bara ändra hur teamen arbetade tillsammans, det krävdes systematiska förändringar av organisationen, det krävdes av vi rev ner alla dessa små och stora pyramider som stod ivägen.

När jag började som utvecklingschef där hösten 2007 var det på en tämligen konventionell tjänst, med en mix av det som brukar ingå i rollen som chef över en grupp experter i en teknikdriven organisation: Stort förväntat inflytande och involvering i teknikval och arkitektur, en ganska stor dos av "leda och fördela arbetet" tolkad i bred mening, och så som sidoverksamhet, det mer tra-

ditionella linjechefsansvaret för tider, personal, rapportering och koordinering med andra avdelningar.



Klassisk hierarkisk organisation

Fem år senare, när jag bestämt mig för att gå vidare, och i stället arbeta som coach och hjälpa andra företag och organisationer att uppnå det vi lyckats med skrev jag ett öppet brev till min efterträdare:

Om några veckor avslutar jag min tjänst som utvecklingschef. Normalt sett brukar ju en avgående chef ersättas med en ny. Jag är emellertid inte så säker på att det är en bra idé, åtminstone inte med någon som förväntar sig att få bestämma över så mycket. Jag har nämligen, kan jag så här i efterhand se, delvis ägnat mig åt att dekonstruera den traditionella utvecklingschefsrollen.

Det här är historien om den förändringen, om vår revolution inifrån.

Agila manifestet

a>Vintern 2001 samlades 17 ledande experter inom mjukvaruutveckling under två dagar i skidorten Snowbird i Utah, USA. Alla hade under det senaste decenniet på olika sätt försökt finna nya sätt att organisera mjukvaruutveckling, och flera av dem hade utvecklat egna metodologier, så som den ovan nämnda Scrum av Jeff Sutherland och Ken Schwaber, eller eXtreme Programming av Kent Beck.

På den tiden brukade man tala om dessa metoder som "lätteviktiga", men, om historien är korrekt, tyckte en av deltagarna att han inte vill befinna sig på någon bantningsretreat, och föreslog i stället ordet "agile", amerikanska för lätttrörlig. När de lämnade Snowbird hade de lyckats formulera något de kallade det "agila manifestet". I manifestet slår man fast fyra grundvärden som man menar lägger grunden för ett modernt och flexibelt sätt att arbeta:

1. *Individer och interaktioner* framför processer och verktyg
2. *Fungerande programvara* framför omfattande dokumentation
3. *Kundsamarbete* framför kontraktsförhandling
4. *Anpassning till förändring* framför att följa en plan

Som ett tillägg formulerade man också 12 mer konkreta principer:

1. Vår högsta prioritet är att tillfredsställa kunden genom tidig och kontinuerlig leverans av värdefull programvara.
2. Välkomna förändrade krav, även sent under utvecklingen. Agila metoder utnyttjar förändring till kundens konkurrensfördel.
3. Leverera fungerande programvara ofta, med ett par veckors till ett par månaders mellanrum, ju oftare desto bättre.
4. Verksamhetskunniga och utvecklare måste arbeta tillsammans dagligen under hela projektet.
5. Bygg projekt kring motiverade individer. Ge dem den miljö och det stöd de behöver och lita på att de får jobbet gjort.
6. Kommunikation ansikte mot ansikte är det bästa och effektivaste sättet att förmedla information, både till och inom utvecklingsteamet.

7. Fungerande programvara är främsta måttet på framsteg.
8. Agila metoder verkar för uthållighet. Sponsorer, utvecklare och användare skall kunna hålla jämn utvecklingstakt under obegränsad tid.
9. Kontinuerlig uppmärksamhet på förstklassig teknik och bra design stärker anpassningsförmågan.
10. Enkelhet – konsten att maximera mängden arbete som inte görs – är grundläggande.
11. Bäst arkitektur, krav och design växer fram med självorganiserande team.
12. Med jämna mellanrum reflekterar teamet över hur det kan bli mer effektivt och justerar sitt beteende därefter.

För en icke-utvecklare kan både värdena och principerna möjligen kännas både för konkreta och lite främmande. I mjukvaruvärlden har dock det agila manifestet, dess värden och principer - och mer specifikt de olika metoder som man kan finna under det vi kan kalla det agila paraplyet - fått ett enormt inflytande. Faktum är att det knappt finns några mjukvaruutvecklande företag kvar som inte på något sätt anpassat sitt arbete efter principerna i agila manifestet. Agila metoder, så som Scrum eller så kallade Kanban-inspirerade sätt att arbeta, är i dag vardagsmat inom mjukvaruutvecklande organisationen.

Det är heller inte särskilt svårt att bredda innebörden av både värderformuleringarna och principerna så att det blir mer generella och tillämpbara långt utanför just programmeraryrket. Idag blandas principerna från agila manifestet men många andra organisationsfilosofier, så som Lean eller systemtänkande och sprids till företag långt utanför mjukvarubranschen.