

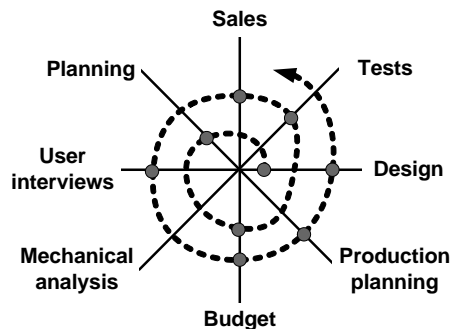


Käre läsare,

Gensvaret från dig som läsare har varit väldigt positivt till de två första veckobreven. Därför fortsätter vi på den inslagna vägen, med den skillnaden att veckobrevsnumreringen får följa veckonumret.

## Växla mellan olika arbetsuppgifter—var inte beroende av en specifik förutbestämd arbetsordning

Din kreativa förmåga ökar om du växlar mellan olika arbetsuppgifter. Tempo, initiativ och pengar går förlorade om människor sitter och väntar. Om du av någon anledning inte kan fortsätta med det du håller på med, så växla över till det som är det näst viktigaste tills dess att du kan återuppta det du ursprungligen höll på med. Ny forskning har nämligen visat att ju erfarenare och skickligare en produktutvecklare är, desto mer itererar de mellan olika arbetsuppgifter.



Till skillnad mot andra PD-teorier rekommenderas i DPD att man hoppar mellan olika arbetsuppgifter i avsikt att höja den kreativa förmågan.

Att slaviskt följa en checklista betyder alltid en större eller mindre tidsförlust. Däremot kan det vara bra att konsultera checklisten efteråt för att se att man inte missat något väsentligt.

I DPD säger man att man i varje ögonblick skall göra det för stunden viktigaste och inte det som går så det näst viktigaste, osv. Vad är då det viktigaste? Jo, enligt DPD är det viktigaste det som för mot målet, förverkligar visionen. I LPD ger man en snarlik förklaring; man skall utnyttja tiden för att skapa kundvärde eller nyttigt vetande.

Avsikten med ovanstående regel är att skapa effektivt resursutnyttjande.

Det gäller också Morgan-Likers Princip 3: "Create a leveled PD process flow". Men här anlägger man ett utifrån och uppifrån perspektiv. Det är någon (högre chef kan man förmoda) som skall skapa ett jämnt flöde i processen. För att det skall fungera måste den som designar processen och dess flöde ha perfekt kunskap, som naturligtvis inte finns. Resursutnyttjandet kommer därför aldrig att bli riktigt effektivt med ett sådant perspektiv.

Detta söker man lösa i DPD genom att utnyttja en "emergent" effekt, som är ett begrepp från komplexitetsteorin (Googla på: complexity theory/science, eller kolla på [www.complexityforum.com](http://www.complexityforum.com) ).

Enklast är att betrakta en stor fågelflock (eller fiskstim) som graciöst undviker hinder och i böljande flykt far fram över himlen. Längre trodde man att det fanns en ledarfågel som de andra följde. Helt enkelt därför att det var så man såg människoskapade organisationer. Chefen var organisationens huvud (head of the department) som tänkte och de underställda var kroppens lemmar som utförde arbetet.

Men så fungerar inte fågelflocken. Flocken och dess beteende bygger på tre enkla regler som upprepas och följs av varje flockmedlem. De är 1/ flyg inte in i något objekt, 2/ styr mot tyngdpunkten av de fåglar som är närmast, och 3/ ha samma fart som genomsnittet av de fåglar som är närmast. Dessa regler upptäcktes 1987 av Craig Reynolds och användes av Disneys tecknare vid datoranimeringen av en million fladdermöss som lämnade en grotta. Det såg mycket naturtroget ut.

Vad kan vi nu lära av detta? Kanske främst två saker: 1/ nyare forskning har visat att denna typ av ”bottom-up”-organisering ger komplexa system som är effektivare än varje ”top-down”-designat system, och 2/ om vi vill nå verklig effekt med våra LPD-ansträngningar så måste vi komma åt och påverka ingenjören i hans/hennes dagliga arbete med enkla regler av typen: ”Växla mellan olika arbetsuppgifter—var inte beroende av en specifik förutbestämd arbetsordning”.

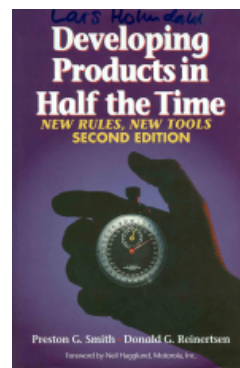
## Veckans bok

**S**kall du bara köpa en bok om produktutveckling, så köp den här. Så sa jag för 11 år sedan när jag första gången läste den här boken och det håller nog faktiskt ännu.

Det är helt enkelt en mycket bra bok som på 290 sidor får med det mesta om produktutveckling från identifierad affärsmöjlighet, ekonomiska analyser, strategier och metoder, teambuilding, organisering, processer, styrning, risk management, över högsta ledningens betydelse fram till och med produktion.

Språket är klart, ämnet väl strukturerat och texten är informationsrik. Jag har bara två invändningar, på sidan 55 säger Preston/Reinertsen att man inte regelmässigt kan produktutveckla snabbt samtidigt som man utvecklar ny teknik—det kan man visst om man använder Dynamisk Produktutveckling (DPD), och vidare på sidorna 243-49 avhandlas concurrent engineering (CE) inte helt korrekt enligt gängse synsätt.

Men totalt är ändå det här en av de bästa böcker du kan köpa. Boken får ★★★★★.



Preston G. Smith  
Donald G. Reinertsen  
**Developing Products  
in Half the Time**  
2<sup>nd</sup> ed., John Wiley &  
Sons, Inc, 1995  
ISBN 0-471-29252-4