

- Detta är Extreme Programming
- XPs tolv beprövade grepp
- Expertkommentarer om XP
- Vilka verktyg behövs?
- XP drar fulla hus
- Erfarenheter från företag
- Slutsatser om XP användning



## Extreme Programming

– Programvara får kvalitet – när utvecklarna själva får välja

Är din verksamhet beroende av programvara av god kvalitet?

Allt fler är helt beroende av omfattande programvara för att produkter och organisation skall fungera.

Att utveckling av programvara ofta innebär stora risker är välkänt. Budgetutfall och kvalitet varierar, minst sagt. Varför blir inte programvaran bättre? Är det utvecklarens eller organisationens fel?

Erik Lundh, verksamhetsutvecklare i programvaruindustrin, rapporterar om Extreme Programming, XP, som ger utvecklare bättre medel för kvalitet.



*Svalkande eftersnack. Ivrigt diskuterande Kent Beck m fl frontfigurer inom XP, glömmet bort att bada i Sardiniens salta vatten på XP2000-konferensens sista kväll.*

### Lyhörda självorganiserande team garanterar kvaliteten

Nu tar utvecklarna saken i egna händer. Extreme Programming, XP, har fått häpnadsväckande genomslag bland erfarna utvecklare. XP riktar sig direkt till de som gör jobbet. XP blandar avancerad teknik och stenhårt disciplinerad process med snabb feedback. Utvecklarna, idag en bristvara, ställer upp på den tuffa arbetsordningen då de som team ser chansen att få resultat och kvalitet de kan vara stolta över. Men resten av organisationen måste bli lika bra annars blir obalansen akut.

IBM, HP, Nokia, Motorola, Ericsson med flera ser, efter många år av tunga kvalitetssatsningar på

programvara, möjligheten att genom XP ge den stora gruppen medarbetare, utvecklarna, chansen att själva garantera kvaliteten i programvaran. Flertalet företag arbetar sedan länge med delar av XP, då allt är kända metoder som kombinerats ihop så att de kompenserar varandras brister.

XP handlar om självorganiserande team som arbetar oerhört lyhört för beställarens önskemål med korta cykler och ständig uppdatering av prioriteter. Att kravspecifikationen förändras med beställarens förståelse är ett normalt inslag i XP. För att bli klara i tid måste många programvaruprojekt påbörjas innan kravbilderna är klar. Exempelvis

måste produkter som baseras på nya standarder utvecklas medan standarden fastställs.

## Detta är Extreme Programming

Extreme Programming, XP, är en hårt disciplinerad lättviktsprocess för programvaruutveckling som betonar teamwork, kommunikation, feedback, enkelhet och problemlösning. Deltagarna i teamet upplever en entydig stark känsla i process och samhörighet, "flow". XP fokuserar på att få ett team att fungera med ett begränsat antal medlemmar. Teamets storlek begränsas bland annat av att behovet av skriftlig kommunikation och tillfälliga dokument tenderar att växa exponentiellt i större grupper. Med begreppet lättviktsprocess menas främst en process med låg overhead i förhållande till i slutprodukten värdeskapande tid och energi. XP skapades för att hjälpa ett mindre team att snabbt utveckla stabil programvara i projekt där kravbilderna ofta snabbt ändras under projektets gång.

VISA betalningssystem skapades i slutet av 60-talet med självorganiserande lättviktsprocesser som var styrda av idéer och värderingar. Tre-tusen konkurrerande amerikanska banker förenades under VISA-systemets första år. Läget i kritiska projekt markerades av en odiskad kaffe-kopp som flyttades längs de whiteboards som utgjorde planen.

Karakteristiskt för lättviktsprocesser är just det starka inslaget av gemensamma värderingar och ideal som ersätter tusentals detaljinstruktioner. Idealet är att befinna sig i ett ordnat tillstånd på gränsen till kaos. Då gynnas syftet med projektet maximalt. Dee Hock, plastkortets skapare och grundare av VISA,



arbetar idag med att hjälpa ideella organisationer att minska byråkratin med synsätt som leder till lättviktsprocesser.

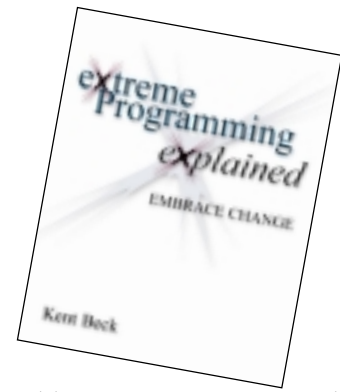
XP växte fram, under Kent Becks ledning, ur design patterns-rörelsen med inslag av refactoring och influerat av utvecklingsmiljöer som Smalltalks främsta fördelar. Kent Beck presenterade XP på OOPSLA som "Turn all dials up to Ten!". D v s att göra ett antal bra saker så mycket det bara går. Enkel design var bra, vrid upp till max. Att putsa på koden var bra, vrid upp till max. Att bygga produkt ofta var bra, vrid upp till max. Beck testade fram ett dussin bra grepp som kombinerar tekniska, organisatoriska och sociala insikter. De flesta tekniska metoder testade han först på sig själv.

Kent Beck förenar i XP en samling Best Practices som kompletterar varandra. En viktig skillnad mot liknande processer eller metoder är det starka inslaget av konkreta tekniska lösningar som integrerat stödjer XP. Det tilltalar teknikern i många programmerare. Processororientering har ofta, i grunden, handlat om att anta ett managementperspektiv, vilket inte varit lika attraktivt för alla medarbetare. Men genom att även tala till det tekniska intresset hos utvecklare skapar Beck brett stöd i hela teamet för en hårt disciplinerad process.

## Extreme Programming Explained

Kent Becks föredömligt tunna och lättlästa bok Extreme Programming Explained presenterar XP på ett lättfattligt sätt. Bokens viktigaste lärdom kan vara att rätt attityd och stort engagemang (men utan övertid) ersätter mycket av gängse styrmekanismer, vilka enligt Beck härstammar från ett otidsenligt Tayloristiskt synsätt.

Företagsledningen kan vid en yttlig betraktelse avfärda XP som en invitation till hackers. Dock visar erfarenheter från bl a SPIN-SYD, ett nätverk med 30-talet sydsvenska programvaruföretag, att XP först uppmärksammas och omtalas av de mest erfarna utvecklarna. Kent Beck ser, med blandade känslor, att XP i USA används som ett attraktivt lockbete för rekrytering av erfarna



utvecklare, inte minst i Silicon Valley och Seattle.

## XP:s tolv beprövade grepp

Enligt Beck är XP:s tolv beprövade metoder, "best practices", en bra uppställning när man står på startlinjen att börja tillämpa XP. Dessa metoder och aktiviteter bygger snabbt insikt och erfarenhet hos utvecklarna inom nyckelområden så att de efter hand kan anpassa verktyglådan efter behov. Om man efter något år fortfarande använder allt i ursprungligt skick anser dock Beck att man verkligen inte arbetar i XP:s anda. XP syftar till att team och individ, då erfarenheten växer, strävar efter att optimera/skräddarsy process och metoder efter projektens behov. En slående likhet med strävan på CMM nivå 5, den optimerande nivån.

### Extreme Programmings tolv beprövade grepp – sammanfattning:

Vissa grepp kan man ta som individ, utan att få med hela teamet. Vissa kräver samförstånd med teamet, vissa kräver att alla i teamet ställer upp. Det framgår under varje rubrik.

- **Metaforen är affärsidén för det vi skall bygga? – "Metaphor"**  
*Påverkar hela teamet då det handlar om att sträva mot samma mål.*
- **Rak, enkel planering – "The Planning Game"**  
*Påverkar hela teamet.*
- **Tidiga, täta produktreleaser – "Small Releases"**  
*Kan påverka hela teamet. En individ kan dock leverera ofta.*
- **Enklast möjliga lösning – "Simple Design"**  
*Kan praktiseras individuellt.*

- **Skriv testkod först av allt**  
– **Testa ständigt** - "TestFirst"

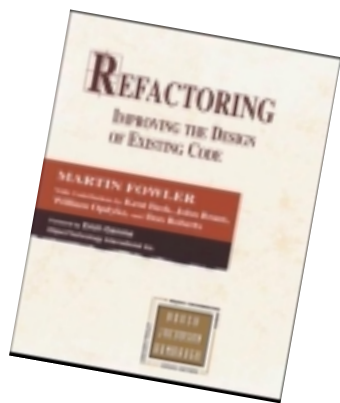
*Kan praktiseras individuellt*

### **Test Infected: Utvecklare lär sig älska att skriva tester!**

Erich Gamma och Kent Beck skriver i artikeln "TestInfected: Programmers Love Writing Tests": "Varje utvecklare vet att de bör skriva tester för programkod. Få gör det. Det vanligaste svaret på frågan 'Varför inte?' är 'Jag hinner inte just nu.' Detta blir snabbt en ond cirkel där man testat allt mindre då tidspressen ökar. Men ju mindre man testat, desto instabilare blir koden och därmed sjunker produktiviteten. Ju mer osäker och oskärt man känner sig, ju mer tidspress upplever man. Utvecklare bränns lätt ut av denna onda cirkel".

- **Ständig förbättring av koden** – "Refactoring".

*Kan praktiseras individuellt.*



Refactoring innebär att omarbete programkod med bibehållet yttre observerbart beteende. En grundbult för att detta skall vara rimligt är effektiv versionshantering av programkoden med täta incheckningar. Refactoring förfinar designen inkrementellt. Ställer höga krav på konfigurationshantering.

### **Några XP-värderingar om programkod och struktur:**

Ett och endast ett ställe – Kod skall ej vara dubblerad. Håll det enkelt – Enklast design vinner.

Ändamålsenlig men löst kopplad kod. (Stödjer TestFirst)

Refactoring är aktiviteten då man "snyggar upp" och lever upp till dessa värderingar.

- **All kod skrivs i par**  
– "Pair programming"

*Kan praktiseras individuellt.*

XP-utvecklare skriver all kod som ingår i slutresultatet i par. Märk väl, två utvecklare på ett tangentbord! Utvecklare bör ha egen arbetsplats för personliga och explorativa moment. Men all "skarp" kod skrivs i par.

- **Alla äger koden**  
– "Collective ownership"

*Kan praktiseras individuellt.*

All kod ägs gemensamt. Alla kan ändra överallt. Detta kräver givetvis kraftfull, rigorös versionshantering och konfigurationsstyrning.

Med rätt stöd på plats kan ändringar som slår i flera moduler utföras mycket snabbt. Friheten att göra ändringar i flera moduler stöds dessutom av TestFirst, Refactoring och Coding Standards.

- **Bygg hela tiden**  
– "Continuous Integration"

*Påverkar hela teamet.*

"Daily Build is for whimps" brukar Kent Beck säga när han talar om "continuous integration"

XP-team bygger hela sin produkt många gånger om dagen.

Det behövs för att alla testfall skall kunna köras kontinuerligt och att alla skall kunna arbeta parallellt. Behovet av högt automatiserad konfigurationshantering är uppenbart.

- **Ingen overtid** – "40-hour Week"

*Kan påverka hela teamet.*

Trötta utvecklare gör misstag. XP-team jobbar inte onödig overtid. Vettigare att sköta hälsan, må bra och vara effektiv när man jobbar.

Att föreskriva 40-timmarsvecka är mer kontroversiellt i USA. Men den sociala regeln inom XP-team säger att man inte jobbar overtid under två veckor i rad.

- **"Kunden på plats" är med i teamet** – "On-site Customer"

*Påverkar hela teamet.*

Ett XP-projekt är oerhört beroende av feedback. Därför sitter en kundrepresentant ("Kunden på plats") som är domänexpert med i teamet. Dvs en person som vet så mycket som möjligt om hur produkten skall fungera i sin tänkta tillämpning.

Kunden på plats skriver krav, prioriterar och svarar på frågor från utvecklarna. Kundens representant sitter i teamet hela tiden men kan jobba med sina övriga arbetsuppgifter då teamet inte behöver input eller feedback.

Man eliminerar tidsödande och kostsamma arbetsdokument mellan kund och team som ej har något värde när produkten är klar. "Kunden på plats" får dock ett mycket större ansvar att styra rätt.

- **Kodningsregler för frihet**  
– "Coding Standards"

*Kan påverka hela teamet.*

För att alla skall kunna hitta rätt i den gemensamma koden krävs en ordentlig kodningsstandard. I parprogrammering turas man om att koda. Då gäller det att ha klara regler och ett slags enkelhetens estetik som ger rak och lättförståelig kod där man inom paret kan avlösa varandra vid tangentbordet "mitt i en mening". Regelkontroll kan automatiseras med verktyg sprungna ur kompilator teknik.

### **Mätningar i XP**

I ett XP-projekt mäter man 5 variabler: Kostnad, Tid, Kvalitet (både teknik och kundtillfredsställelse), Omfattning samt Hastighet. Hastighet är en kvot mellan total arbetstid och arbetstid som gjort nytta för produkten.

### **Några andra lättviktsprocesser**

ASD – Adaptive Software Development (Jim Highsmith)

CRYSTAL LIGHT – Alistair Cockburns metodmatris där man själv väljer ut en metodvariant som passar projektet

DSDM – [www.dsdm.org](http://www.dsdm.org). Påminner i sociala delen mycket om XPs. Saknar dock XPs attraktiva tekniska inslag.

FDD – Feature Driven Design (Peter Coad)

OSS – Open Source Software

SCRUM – Varje iteration är en tjurusing med kaos i hälarna. Mellan iterationerna återfinns man en viss ordning. [www.controlchaos.com](http://www.controlchaos.com)

## Experter kommenterar XP

Tom DeMarco, en av författarna till den banbrytande boken "Peopleware" säger:

–XP är det mest betydelsefulla som händer inom programvaruområdet just nu. Kent Beck kan få större betydelse i detta skede än vad Watts Humphrey och CMM fick i det skede vi just lämnat.

Han oroar sig dock för arbetsmiljön i XP och önskar att folk som arbetar på samma sätt som i Chryslers C3-projekt, i ett enda stort rum, också har enskilda rum där man kan få vara ifred.

Alistair Cockburn konstaterar att det krävs tre förutsättningar för att XP skall bli framgångsrikt:

- låg kostnad att ändra i gränssnitt
- tät kommunikation
- automatiserad regressionstestning.

Beck refererar till "Crossing the chasm" och säger att det stora steget för XP blir att bli accepterat av den stora massan företag. Då måste XP ses som ett rimligt riskalternativ och behöver en etablerad image.

### Varför kallas det XP?

På frågan om hur han valde namnet extreme programming svarade Beck skämtsamt att han strävade efter ett namn som tunga metodgurus som Grady Booch aldrig skulle kunna ta i sin mun.

Dock har Rational, där Booch är en frontfigur, annonserat att man kompletterar Rational Unified Process, RUP för att kunna innefatta även XP. (Rational strävar efter att täcka in de flesta moderna metoder med RUP, t ex DSDM).

### Det tyngre perspektivet

Watts Humphrey, CMM:s urfader, gillar XP! Humphrey uttalar sig oväntat positivt om XP i ett debattinlägg för IEEE. Han påpekar visserligen att Kent Beck inte stödjer sig på insamlade data i sin ursprungliga XP-bok, men säger samtidigt att Humphreys egen forskning kring team-processer, TSP, har samlat data som ger starkt stöd för XP. Invändningarna mot XP gäller främst att XP i sin ursprungliga form inte skalar upp till team större än 10-15 personer, samt att XP inte är beskrivet i detalj. Vilket är helt korrekt.

### XP kan leda direkt till CMM nivå 4

Humphrey verkar hos Software Engineering Institute, SEI som "äger" och utvecklar CMM-standarden. En av hans kollegor Mark Paulk gör gällande, i ett annat debattinlägg för IEEE, att ett XP-team direkt skulle kunna certifieras på CMM nivå 4 om man kompletterar XP med vissa mätningar och införlivar resultaten i feedback-styrningen som redan finns i XP.

### XP möter CMM i Indien

Vi hoppas att Kent Beck och Alistair Cockburn får möjlighet att studera indiska CMM5-team via svenska SPIN-samarbetet med Bangalore i Indien.

Hypotes: En del av Indiens framgångar med CMM kan härledas ur att indiska team naturligt agerar i XPs anda. XP kan därmed ge team i väst samma samarbetsformer som kulturellt präglas i Indien."

(Se även VIs nyhetsbrev Systemkonstruktion i programvara nr 2 år 2000.)

## Vilka verktyg behövs för XP?

Ritverktyg med begränsad kodgenerering ratas till förmån för generella, bättre ritverktyg som Visio. Två gånger under konferensen XP2000 hölls handuppräckning om vilka som föredrog ett generellt ritverktyg för UML-diagram som Visio framför Rational Rose och liknande.

Visio vann stort båda gångerna. En av de som uppmanade till handuppräckning, och som själv röstade för Visio, hör till de ursprungliga utvecklarna av Rational Rose.

Alltför många användare sägs undvika kodgenerering, och använder CASE-verktyg som extremt dyra ritlådor. Det generella ritprogrammet Visio har funnit sedan början av 90-talet. Inriktar sig främst på schematisk linjgrafik som organisations-schema, flödesdiagram, eller som en parentes UML-diagram. Då de generella egenskaperna för strukturerad grafik i Visio drivs av en långt större användarbas, är ritegenskaperna överlägsna samtliga CASE-verktyg på marknaden. Microsoft köpte Visio 1999.

Inom XP uppmuntras däremot användning av konfigurationsverktyg, versionshantering och testauto-

mation. Avancerade verktyg för semantiskt korrekt kodförbättring genom refactoring är också favoriter. Dock finns idag kvalificerade refactoring-verktyg enbart i Smalltalk-miljö.

## XP drar fulla hus

På XP2000, den första internationella konferensen, fanns delegater från HP, Ericsson, Nokia, Nortel, IBM, Motorola, Daimler Chrysler, Siemens, dessutom indirekt Microsoft. Man samlades i slutet på juni i år på Sardinien för den första internationella konferensen om XP: "Extreme Programming and Flexible Processes in Software Engineering - XP2000".

Den som följt mer avancerad litteratur och debatt inom XP, software engineering och computer science, fick sitt lystmäte av kändisar: Kent Beck, Erich Gamma; Martin Fowler, Ralph Johnson, Robert Martin, Don Wells, Ron Jeffreis, Alistair Cockburn. Mycket av XP kommer från den design patterns community som växte upp sedan boken Design Patterns av Erich Gamma med flera kom ut 1995.

Intressant var bland annat erfarenheter av att XP accepteras lättare än andra processer av team med forskarbakgrund.

De som fruktade ett fanatiker-möte om XP blev lättade. De flesta som var där hade infört XP helt eller delvis på ett pragmatiskt sätt. Vilket är helt i XPs anda att odla djupare förståelse för teknik och process. Dave Thomas från IBM, Martin Fowler m fl profiler uttryckte sin glädje att finna att XP-folket verkligen lever som de lär, flexibelt och med respekt för andras idéer.

Intresset för XP är extremt stort. När universitetet i Calgary planerade för konferensen på Sardinien räknade man med ca 60 internationella gäster.

Det kom 161 delegater från hela världen, varav 15 från Sverige. XP-2000 gav mersmak och nästa års konferens presenteras redan på [www.xp2001.org](http://www.xp2001.org)

## Erfarenheter från företag

Motorolas Ron Crocker publicerade på OOPSLA2000 ett papper om erfarenheter av XP som meta-

modell för att samordna utvecklingsteam på skilda platser i världen. I Motorolas modell motsvas individen i ett vanligt XP-team av ett platsteam. Varje platsteam kan välja sina egna interna arbetsmetoder. Samordningen sker dock enligt XP-principer. Intressant var också att den legendariska Motorola-enheten i Bangalore, första civila verksamhet i världen på CMM nivå 5 1993, deltog i detta stora meta-XP projekt.

En av delegaterna på XP2000 i juni var Henrik Warne, Ericsson Radio Systems, Kista. Hur långt har man där anammat XP nu i höst?

– Vi har provat delar av XP efter hand och jobbar med programvara för nya generationer av mobiltelefonens basstationer. Testförfarandet med kontinuerlig testning har fått omedelbart genomslag hos oss. Vi utvecklar i Java och kan direkt använda Kent Becks testramverk Junit ”på originalspråket”. Våra utvecklare regressionstestar hela tiden med ”ett tryck på knappen” som kör växande gemensamma testserier där alla kan stoppa in nya tester.

Parprogrammering har varit mycket givande för de flesta som velat pröva. Sitter man vid tangentbordet, så kan man ställa frågor rätt ut i luften och kollegan är direkt med utan att man behöver förklara. Management har varit positiv till parjobb då man ser det som kompetensspridning samt för nya utvecklare ett slags positiv lärlingsprocess. Ökad mantid har inte oroat i den ekvationen. Utvecklare med inbiten vana att jobba själv kan däremot vara negativa innan de har provat.

Vi tittar just nu på att tillämpa snabb kontinuerlig kravprioritering enligt XP i vissa projekt.

Det vi inte tror på i XP är Becks ”Open Workspace” i bemärkelsen att ständigt ha teamet i ett enda stort rum fullt med datorer som Chryslers XP-projekt C3.

## Slutsatser om XP användning

### Boehms Cost of Change gäller men effekten minskas av XP

Barry Boehms klassiska modell säger att det alltid är billigast att hitta fel så tidigt som möjligt. Kurvan över kostnad för att avhjälpa ett fel steg exponentiellt ju längre produkten kom i processen. Nackdelen är

om man inte kan föreställa sig hur det ska fungera och designfasen drar ut på tiden.

Beck hävdar i sin XP-bok att XP plattar till den exponentiella kurvan. Cockburn hävdar att XP ger effekten att man alltid befinner sig i början av exponential-kurvan.

### Tekniska och ekonomiska faktorer bäddar för XP

XP blir möjligt bland annat tack vare kraftfulla billiga datorer och tillkomsten av avancerade utvecklingsverktyg som säljs i massupplaga. Då kan en standard PC eller laptop innehålla verktyg som tidigare krävde, med dåtidens mått, kraftfulla dedicerade Unix-servers.

Utveckling inom software engineering med exempelvis design patterns, kompilator teknik och nätverksorientering verkar för att det skall vara tekniskt möjligt att kunna hålla fokus i koncentrerade releaser av programkod.

### XP förbättrar inte hela organisationen

XP ger, rätt tillämpat, ett team gynnsamt beteende för programvaruutveckling. Det gör dock inget för resten av organisationen. Är XP den enda programvaruprocessen, krävs hela tiden stor förståelse på individnivå från resten av organisationen. Det är en del av orsaken till att XP kan uppfattas som cowboyverksamhet. Chryslers C3-projekt som var ett tidigt större XP-projekt för löneadministration blev till slut omöjligt att fortsätta av skäl som inte teamet kunde påverka. Resten av organisationen arbetade på en helt annan mognadsnivå när det gällde programvaruutveckling. Det management som startat projektet hade gått till andra uppgifter. Interna konflikter inom Chrysler ledde till att C3-projektet lades ner. XP blev en tid ett internt slagträ inom Chrysler.

Vi tror att CMM eller motsvarande mognadsprocess bör tillämpas på hela organisationen med fokus på management medan XP tillämpas på utvecklingsteam.

XP kan få den stora personalgruppen utvecklare att fungera bra i en organisation som på managementnivå mognats med CMM. CMM tar ett helhetsperspektiv som tilltalar management, medan XP tilltalar utvecklarna. När XP och

CMM samverkar på rätt sätt kan vi få verkligt snabb mognad av stora programvaruutvecklande organisationer.

### Är XP lättare för det mindre företaget?

Av många skäl, inte minst politiska, kan det vara lättare för team i det mindre företaget att anamma XP på en gång i ett eller flera projekt. I större företag kan man försöka introducera utvalda delar stegvis. Detta ökar dock risken för misslyckande då flertalet av XPs olika metoder har misslyckats på egen hand. Det är just interaktionen mellan de olika fördelarna hos olika metoder som kompenserar för nackdelarna hos enskilda metoder. I de flesta fall elimineras helt nackdelarna med andra stödjande metoder, i enstaka fall överträffar fördelarna för helheten metodens nackdelar. Därför kan ett projekt som tillämpar XP fullt ut utnyttja fördelar som man går miste om när man tillämpar enstaka metoder.

### XP driver teknisk och social kompetens

Många har upplevt, eller hört talas om det skräckprojekt där varje utvecklare har begränsad kompetens och bara kan ”vara sitt enda kapitel i grundboken”, särskilt i de senaste årens ”webbfrenesi” samt i många kundspecifika projekt bemannade med resurskonsulter. Beck har mycket erfarenhet av kundspecifik utveckling med blandade team. Därför har XP designats för att driva kompetensspridning och kommunikation. Medlemmar i ett XP-team delar naturligt kunskap genom parprogrammering och kollektivt ägande.

Den utvecklare som trivs i ett XP-team får därmed på köpet en bred personlig kompetensutveckling.

Många upptäcker att även de mest erfarna får nya insikter av att arbeta med mindre erfarna. I parprogrammering blir detta särskilt tydligt.

XP är starkt feedbackorienterat och därigenom i sig en lärande process. Det smittar av sig.

### Utvecklarna kräver bättre arbetsformer!

Martin Fowler sade i sitt avslutningstal på konferensen XP2000:

## Beställning

Följande i företaget önskar också abonnera (gratis) på Nyhetsbrev:

Namn \_\_\_\_\_

Avd \_\_\_\_\_

Företag \_\_\_\_\_

Adress \_\_\_\_\_

Postadress \_\_\_\_\_

Systemkonstruktion  
i programvara nr 3, 2000

# B

Porto betalt  
Sverige

## Ny adress?

Skriv din nya adress och skicka den med den gamla till:  
VI, Sveriges Verkstadsindustrier, Box 5510, 114 85 Stockholm, telefax 08-782 09 00

Jag vill istället ha nyhetsbrevet via min E-post!

Epostadress: \_\_\_\_\_

"If you can't change your organisation – Change your organization. This market is no time to be unhappy with your job!"

Kent Beck tar avstånd från managementkonsulter och vill själv gärna ses som en duktig utvecklare bland andra. Men XP handlar egentligen om empowerment, de senaste årens trend bland managementkonsulter. (Empowerment i någon form blir trendmässigt ofta populärt när efterfrågan på kompetent personal vida överstiger tillgången.)

## Runt om i världen skapas platser i XPs anda

Kent Beck planerar ett utvecklingscenter med inriktning på forskning med kollaborativa utvecklingsprocesser i Oregon, USA där företag kan sätta team att arbeta i XPs anda.

Beck blev under XP2000-konferensen intresserad av ett liknande projekt i Sverige (PSC Helsingborg) som Erik Lundh är engagerad i sedan två år tillbaka.

Ken Auer, en annan av XPs frontfigurer, håller i USA på att bygga upp en "XP Software Studio" med positiva drag av konstskola.

## Mer att läsa

Detta nyhetsbrev är en nedkortad version av Erik Lundhs artikel om XP. Läs mer om:

- XPs tolv beprövade grepp med exempel på testning
- Kent Beck på turné i Sverige
- Elva stycken bokreferenser
- Mer om författaren Erik Lundh

Hela artikeln finns att läsa på [www.vi.se](http://www.vi.se) – gå in på teknik/systemkonstruktion i programvara.

## Om författaren

Erik Lundh har arbetat med massprodukter som kräver programvara sedan tidigt 80-tal. Verksamhetsutveckling av programvaruindustrin sker genom nätverk, konsult-, samt styrelseuppdrag.



Mer information:  
[erik.lundh@compelcon.se](mailto:erik.lundh@compelcon.se)

Se nyhetsbrev på  
[www.vi.se](http://www.vi.se)

## Några bokreferenser

ISBN	Titel	Författare	Förlag
0201485672	Refactoring : Improving the Design of Existing Code	Martin Fowler et al	Addison-Wesley
0201616416	eXtreme Programming eXplained : Embrace Change	Kent Beck	Addison-Wesley
9177385268	I en kaordisk tid - Dee Hock plastkorsets skapare och grundare av VISA	Dee Hock	Svenska Förlaget

## Referenser-Webb

[www.extremeprogramming.org](http://www.extremeprogramming.org)

[www.xprogramming.com](http://www.xprogramming.com)

Utmärkt om XP, samlar varianter av xUnit.

[www.c2.com](http://www.c2.com)

Klicka på ExtremeProgramming

Roadmap. Här föddes XP!

[collaboration.csc.ncsu.edu/laurie](http://collaboration.csc.ncsu.edu/laurie)

Prof Laurie Williams har mätt nedlagd tid vid parprogrammering.

[www.junit.org](http://www.junit.org)

[www.xp20001.org](http://www.xp20001.org)

## Nyhetsbrevet Systemkonstruktion i programvara

ISSN 1403-9537

utges av

VI, Sveriges Verkstadsindustrier,  
Box 5510, 114 85 Stockholm,  
tfn 08-782 08 00, fax 782 09 00.

Du kan skicka e-post enligt  
följande: [fornamn.efternamn@vi.se](mailto:fornamn.efternamn@vi.se)  
(obs! utan prickar över å, ä och ö)  
VIs webbplats har följande adress:  
[www.vi.se](http://www.vi.se)

Anställda i VIs medlemsföretag  
kan kostnadsfritt abonnera på  
nyhetsbrevet.

### Ämnesansvarig

Göran Östlund 08-782 08 51  
[goran.ostlund@vi.se](mailto:goran.ostlund@vi.se)

### Sekreterare

Ulla Magnusson 08-782 09 83  
[ulla.magnusson@vi.se](mailto:ulla.magnusson@vi.se)

### Redaktör, ingenjör

Fleming Falkenstad 08-782 08 32  
[fleming.falkenstad@vi.se](mailto:fleming.falkenstad@vi.se)