

Introduktion till eXtreme Programming

1 Introduktion

eXtreme Programming (XP) har uppstått som en reaktion på andra, allför omfattande, processers misslyckande när det gäller att underlätta för projektgruppen att uppnå sina mål. Processen riktar sig i första hand till små projekt. Enligt Kent Beck, som är en av upphovsmännen, kommer ordet extrem i processens namn ifrån att man har tagit bort allting dåligt i processen och enbart behållt det som är bra men i gengäld gör man extremt mycket av de aktiviteterna.

Ledorden för arbetet är kommunikation, enkelhet, återkoppling och mod. XP är en 'lätt' process där man enbart gör de aktiviteter som direkt bidrar till slutproduktens kvalitet, exekverbarhet och förvaltningsbarhet. Arbets sättet och värdegrunden i processen är utformad i syfte att minimera byråkrati och onödigt arbete samt att minimera 'cowboy-coding' genom att betona vikten av ett disciplinerat angreppssätt för test, kommunikation och omarbete.

2 Översikt

Ett XP-projekt arbetar med releaser som tar upp till ett par månader att leverera. Releasen delas in i ett antal två-veckors iterationer, som delas in i ett antal arbetsuppgifter som tar upp till ett par dagar styck. Planering och allokering av krav till releaser och iterationer sker i enlighet med den verklighet som projektet verkar i. Projektet styrs utifrån fyra variabler; kostnad, tid, kvalitet och omfattning. Tanken är att kunden som mest får kontrollera tre av dessa, den fjärde anpassar projektgruppen utifrån vart kunden lägger nivån för de övriga tre.

I princip skulle ett XP-projekt kunna se ut som följer: projektgruppen samlas på morgonen, tillsammans med kunden, och väljer ut dagens historier (de funktioner som skall realiseras) bland de som sitter uppsatta på väggen. Respektive projektmedlem väljer vad han vill göra och letar rätt på en annan person att samarbeta med. Parprogrammering anses höja såväl kvalitet som produktiviteten. Arbetet börjar med att det aktuella paret diskuterar lite grann kring vad de gjort tidigare i projektet, lärdomar de fått och vilken kod de producerat. Efter denna kunskapsöverföring tar de tag i sin 'historia'. De börjar

med att fundera kring testfall och vid behov går de bort till kunden, eller någon annan i projekt, för att klargöra vissa detaljer. När de vet vad och hur koden skall testas, skriver de den och kör testerna. Efter en stund upptäcker kanske en av dem att koden är lite onödigt krånglig och omarbetar delar av den så att strukturen blir lite tydligare. I samband med detta kommer de också på att de med några små ändringar skulle kunna återanvända en komponent som ett annat par tog fram tidigare veckan. De implementerar då snabbt ändringarna i den komponenten, testar den för att säkerställa att allting fortfarande fungerar och fortsätter sedan med sin egen historia. När de känner att de är färdiga integrerar de sin kod med det befintliga systemet och kör igenom samtliga tester för samtliga funktioner. När alla testerna är godkända avslutar de arbetet med sin historia och går tillbaka till samlingslokalen. Där väljer de ut en ny historia och plockar ned den från väggen.

3 Grundläggande koncept

- [Planeringsprocessen](#)
- [Små releaser](#)
- [Metaforer](#)
- [Enkel design](#)
- [Test](#)
- [Omfaktorering](#)
- [Parprogrammering](#)
- [Gemensamt ägande](#)
- [Kontinuerligt integrationsarbete](#)
- [40-timmars vecka](#)
- [Tillång till kunden](#)
- [Kodstandards](#)

Planeringsprocessen (kallas ibland planeringsspelet) - Planeringsarbete enligt XP bygger på att kunden uppskattar verksamhetsnyttan av respektive krav (önskad funktion) och använder sig av kostnadsuppskattningar från utvecklarna för att avgöra vad som skall göras omedelbart och vad som skall skjutas på framtiden. Effekten av detta arbetssätt är att det blir enkelt att styra projektet i mål.

Små releaser - XP-projekt produktionssätter väldigt små och enkla system väldigt snabbt och uppdaterar sedan dessa ofta.

Metaforer - Projektgrupperna i XP-projekt använder ett gemensamt 'namnsystem' och arbetar utifrån en gemensam systembeskrivning (arkitektur) som styr både utvecklingsarbetet och kommunikationsformen. Detta kan till exempel ta sig uttryck som att Chryslers lönehanteringssystem använde sig av klassnamn m.m. från löpande bandet. Genom konsekvent och enhetlig benämning av olika saker underlättas förståelsen för systemet.

Enkel design - Ett system som byggs med hjälp av XP bör vara det enklast tänkbara system som möter de överenskomna kraven. I XP bygger inte projekten för 'framtiden'. I stället ligger fokus på att leverera verksamhetsnytta. Det är dock givetvis viktigt att uppnå en bra design och det säkerställs genom 'refactoring' i XP. Detta diskuteras vidare nedan.

Test - XP-projekt fokuserar konstant på verifiering och validering av systemet. Programmerarna utvecklar mjukvaran genom att skriva testerna först, sedan skriver de koden som uppfyller kraven som återspeglas i testerna. Kunder tillhandahåller acceptanstester som garanterar att systemet får de egenskaper det behöver.

Omfaktorering - XP -projekt förbättrar kontinuerligt systemets design under hela projektet. Det görs genom att hålla koden så ren som möjligt, utan duplicering, enkel men ändå komplett.

Parprogrammering - Programmerarna skriver koden i par om två (dvs två personer arbetar tillsammans framför en maskin). Efterforskning har visat att detta resulterar i bättre system till samma eller lägre kostnad än när programmerarna får sitta ensamma.

Gemensamt ägande - All kod tillhör alla programmerare. Detta ökar tempot i projektet, eftersom så fort någonting behöver ändras så kan vem som helst omedelbart göra ändringen. Projektet slipper därmed onödiga väntetider.

Kontinuerligt integrationsarbete - XP team integrerar och bygger systemet flera gånger per dag. Detta säkerställer att hela projektgruppen hålls synkroniserad och uppdaterad och därmed är det också grunden för ett snabbt framåtskridande. Frekvent integration har också visat sig i det långa loppet eliminera integrationsproblem som ofta dyker upp i projekt som ej integrerat frekvent.

40-timmars veckor - Trötta programmerare gör fler misstag. XP-projekt arbetar inte onödigt mycket övertid vilket håller utvecklarna pigga, friska och effektiva.

Tillgång till kunden - Ett XP-projekt styrs av en i förväg identifierad kund som är bemyndigad att definiera, prioritera och svara på frågor om kraven. Resultatet av detta blir att kommunikationen ökar och den skriftliga dokumentationen minskar.

Kodstandards - För att en projektgrupp skall kunna arbeta effektivt i par och gemensam äga all kod så måste alla programmernas kod se likadan ut. Detta uppnås genom ett regelverk som säkerställer att koden kan kommuniceras på ett entydigt sätt.

4 Mer information

- [White paper om XP](#)
- www.extremeprogramming.org
- www.xprogramming.com
- [http://c2.com/cgi/wiki?Extreme ProgrammingRoadmap](http://c2.com/cgi/wiki?ExtremeProgrammingRoadmap)