

Extreme Programming Explained – En sammanfattning

Extreme Programming är en metodologi för utveckling av programvara och en form av agile software development. XP består av en rad dagligen utförda metoder och principer som är byggda på för XP viktiga värden. Tanken bakom XP är att effektivisera och förenkla mjukvaruprojekt, se till att de håller projektplanen och att kunden blir nöjd med resultatet.

Inom XP är teamet det viktigaste. Mycket viktigare än individuellt skickliga programmera som inte kan samarbeta är medarbetare som lätt kan samarbeta och hjälpa varandra. Ett XP team består inte bara av programmerarna. I teamet bör man få med programmerare, coacher, testare och framförallt kunden. Det är ju kunden som vet vad han vill ha och som kan avgöra om resultatet är lyckat.

Värderingar

XP baseras på fem värderingar som ska guida teamet genom utvecklingen: kommunikation, enkelhet, återkoppling, mod och respekt.

Kommunikation är viktigt för gruppkänslan inom teamet och för effektivt arbete. När ett problem uppstår är det antagligen någon som redan kan lösningen, med kommunikation kan man finna denna någon. Så fort ett problem uppstår bör du fråga dig själv om det beror på dålig kommunikation och vilken kommunikation som krävs för att problemet inte ska uppstå igen. Det är också viktigt att hela tiden kommunicera med kunden. Bestämna vad som ska implementeras härnäst, och återkoppla det som har implementerats.

Försök att alltid implementera den enklaste möjliga lösningen. Man tjänar ingenting på komplicerad kod som inte tillför något extra. Enkelhet är bara relevant i sitt sammanhang: det ska vara den lösning som är simplest för just denna uppgift och detta team.

Under utvecklingen av ett system kommer många saker att förändras. Problemet är inte själva ändringen utan att anpassa sig därefter. För att kunna anpassa sig behöver man feedback. Detta samlas in kontinuerligt under utveckling, testning och körning.

Principer

Vad är effektivast? Långa dokument eller en konversation? Skrift tillåter visserligen en stor publik men saknar talets möjlighet till förtydligande, direkt feedback och brainstorming.

För att binda samman värderingarna med tillämpningarna har XP definierat ett antal principer. Bland dessa kan man lyfta fram följande:

- Man måste tänka på att det är människor som utvecklar kod, se till att uppfylla människornas behov och hitta balansen mellan individens och teamets behov.
- Man bör också tänka på att handla på ett sätt som ger företaget ett värde. Någon måste betala för jobbet och dessa förväntar sig något tillbaka. Alla aktiviteter ska vara givande för samtliga inblandade.
- Gör det bästa du kan idag i strävan att lära dig mer så du kan göra ännu bättre imorgon
- Ett team behöver medlemmar med olika kunskaper, perspektiv och attityder. Man behöver olika idéer för att kunna hitta den bästa.
- Ett bra team jobbar inte bara. Det reflekterar över hur och varför de gör som de gör. Detta krävs för att utvecklas både individuellt och som team.
- Försök att se möjligheter istället för problem. Istället för att bara försöka lösa problemet måste du se chansen att skapa något riktigt bra.

Tillämpningar

För att stötta värderingarna och principerna har XP ett antal tillämpningar:

- Genom att hålla koden enkel blir designen enkel och koden blir lättare att förstå.
- Börja med att skriva testerna till koden. Programmerarna skriver tester som testar kodsnuttar och kunden skriver tester som testar funktionaliteten i stort.
- Ta kontinuerligt bort dubblerad och komplex kod.
- Programmerare arbetar i par och delar dator. Medan den ena skriver granskar och reflekterar den andra. På så vis minskar felen.
- Alla har rätt att ändra i all kod. Koden ägs gemensamt.
- Kunden bör finnas på plats så ofta som möjligt och arbeta så nära utvecklarna som möjligt för att kunna svara på frågor och verifiera att systemet fungerar som tänkt.
- Så fort något nytt implementerats byggs systemet och integreras med den färdiga koden, gärna flera gånger om dagen.
- Programmerare arbetar bättre om de är utvilade. Begränsa arbetstiden till 40 timmar per vecka och tillåt inte övertid flera veckor i rad.

Planering

För varje implementeringsuppgift eller funktion kunden önskar skriver man ett "story card". Ett story card är ett papperskort lagom stort för att kunna skriva en kort beskrivning av funktionen som önskas. Fördelarna med just kort är att det är överskådligt, lättförståeligt, tydligt och har visats vara effektivare än att t.ex. mata in uppgifterna i någon form av datoriserad lösning.

Varje vecka väljer kunden ut de funktioner han vill få färdiga och varje programmeringspar tar sig an en eller ett par kort. Detta utgör veckans planering. När veckan är slut och korten har implementerats får programmerarna en ny uppsättning samtidigt som det de gjorde förra veckan sätts samman, testas och analyseras.

Alla story cards sammanställs på en tavla i den lokal som teamet delar för att alla, inklusive kunden ska se vad som pågår, vad som har gjorts och vad som kommer.

Design

Istället för att börja med att bestämma en design för hela koden och sen inse att det inte fungerar har man i XP som mål att ständigt förbättra designen. Man börjar med en design som är precis så bra att man tar sig vidare och anpassar den sen ständigt efter de kunskaper och erfarenheter man får under projektets gång. Stora designförändringar bör tas i små steg och man bör alltid hålla koden simpel och undvika duplicering. För att hålla koden så enkel som möjligt bör man anpassa den för alla som ska arbeta med den, göra den kommunikativ med väl valda variabel och funktionsnamn, faktorisera den väl och bygga den av så få element som möjligt. Detta ger mindre att testa, dokumentera och kommunicera.

Skalbarhet

Ska man bygga ett större system så behövs ett större team och det blir mer tidskrävande att utföra XP tillämpningarna som vanligt. Lösningen på detta är att dela upp arbetet på flera team.

Börja med att fundera över om det skulle gå att utföra arbetet med ett mindre team. Går inte det så kan du påbörja projektet med ett litet team och sen, när du fått grepp om vad som krävs, dela upp projektet i flera mindre delar och låta varje del skötas av ett mindre team. Det som krävs är bra kommunikation mellan teamen och att integrera alla delar ofta för att se till att de passar samman.

Slutligen

Skjut inte på problemen, kommunicera mera, försök aldrig dölja något och slarva inte.