

Del 2 Uppgift 1-3

1. UML visar tydligt ett förklarande diagram över hur klasser och projektet ska byggas. Som först visar ett klassnamn. Sedan kommer nästa del med attributen till klassen med alla medlemsvariabler som ingår. Sist listas alla metoder som ska ingå i klassen. Fördelen med UML gentemot vanlig källkod är att det ger en bra överblick över koden. En snyggare version om man vill kalla det så, en mer designad kod. Koden visar inte bara hur programmeringen är gjord, vilket får läsaren av koden att inte alltid kunna hänga med i vad som hänt, utan istället visar den mer hur utvecklaren av koden har tänkt ut projektet. Man gör alltså en lista (klass) som man sedan ser tydligt hur utvecklaren har tänkt. På ett strukturerat sätt. Program som utför dessa metoder eller utföranden är mer eller mindre ett UML-program.

2. Att använda redan färdiga lösningar/program kan vara produktivt i vissa fall. En liten summa pengar kan motsvara en stor summa projekttid.

Exempel 1: Man blir kontaktad av ett stort lager som behöver ett nytt system för sin verksamhet gällande inventering, lagervaror, scheman av olika slag och diverse lagerområden. Då kan man använda sig av SAP, istället för att sitta och programmera alltihop kan man på så sätt bespara sig själv eller sitt företag en stor mängd tid.

Exempel 2: Ett mindre småföretag som en liten livsmedelsaffär som behöver ett kassa system eller liknande, kan man istället spara programmeringskostnad genom att ta sig tiden till att skapa allt själv.

Exempel 3: Kanske inget relevant exempel med tanke på att det inte berör själva projekttiden i första hand men har ändå med det att göra. Jag tänker att om man har kommit på en briljant affärsidé eller liknande och man vill få ut den här produkten så snabbt som möjligt på marknaden då man känner att just den här idén kommer kanske dyka upp förr eller senare om man inte gör slag i saken med en gång. Därför väljer man att köpa någon av dessa tjänster för att snabba på processen att få ut produkten så fort som möjligt. Om man ska relatera till sig själv så är det i sådana lägen jag själv hade kunnat tänka mig att använda en sådan tjänst.

3. Thin Client

- * (nackdel) Mycket beroende av internet resurser.
- * (nackdel) Har väldigt begränsade resurser själv.
- * (fördel) Hanterar bäst enkla uppgifter som är lagrade på en nätverks server.

Typiskt för Thin Client är att det ofta används som gränssnitt i affärssystem. Gäller datorer med få lokala program. Innehåller ingen information själv utan utför allt via internet.

Thick Client

- * (fördel) En arbetsstation som inkluderar mestadels alla grundläggande komponenter för att utveckla och verkställa mjukvara applikationer på egen hand.
- * (fördel) Använder inte upp några server resurser när en applikation verkställs.
- * (nackdel) Måste vara kopplad till huvud-servern för att kunna utföra någon nerladdning eller uppdatering av systemet.
- * (fördel) Bra att använda och som används ofta i miljöer där huvud-servern har låg internet hastighet, låg förvaring eller behöver jobba offline.

Rich Client används i stora system, till exempel i Microsoft Excel. Används istället när datorn har många lokala program. Innehåller mycket information.