

## X2

---

2015-04-28

### COBOL

Vi märkte direkt att detta språk var mycket likt assembler, så vi bestämde oss för att lösa det som om det var det vi skrev i även om vi förstår att det möjligen finns bättre metoder. Språket kändes inte som att det bidrog med speciellt mycket, utan mer som ett assembler 2.0.

Fördelen med COBOL var att det var likt assembler. Av denna anledning var det snabbt att komma igång med.

Utifrån vad vi förstår skulle man ifall man ska bygga ett större system i COBOL behöva använda liknande metoder som vid assembler programmering. Det skulle förmodligen var otroligt dåligt för större projekt också.

### Smalltalk

Smalltalk var mycket intressant i och med att man skickade meddelande mellan objekt. Man kunde få syntax att se väldigt naturlig ut. När man väl hade förstått hur det fungerade var det väldigt trevligt att arbeta med. Nackdelen med att skicka meddelanden var att detta var en konstruktion som vi inte var vana vid och det finns nästan inga bra källor för information, vilket gjorde att det tog lång tid för oss att göra implementation för detta lilla projekt. Något som var bra är att det är objekt orienterat på ett så tidigt språk. I ett större projekt tror vi att det kan vara trevligt att arbeta med, även om möjligen kan bli rörigt med labels. Det är defenitivt mycket bättre än COBOL

### Erlang

Erlang är ju underbart! Trots att det är så gammalt känns det så modert. Det känns som att Erlang har tagit en mer pragmatisk approach till funktionell programmering än Haskell. Christoffer använder detta språk som MVK projekt och upplever att det är mycket bra att arbeta med i större projekt. Vi har inte märkt några större nackdelar med språket, men som funktionellt språk så passar det ju inte till alla sorters programmering.

### Reflektion

Man kan tydligt se hur språken utvecklas över tiden. Man startade väldigt nära assembler och har sedan hela tiden gått mot högre abstraktionsnivåer. Det är också väldigt tydligt hur man går mot mindre och mindre rader, vilket vi tror har att göra med att abstraktionsnivån ökat.