

# OOPD uge 1

Kenneth Christensen, Michael Jensen  
Rune Pedersen, Rasmus Hansen

14. maj 2014

# 1 Reviews

## 1.1 No Silver Bullet

"No Silver Bullet - Essence and Accidents of Software Engineering" er en artikel omkring software udvikling, skrevet af Fred Brooks i 1986. Brooks starter ud med at tilsidesætte software udvikling med myten om en varulv. Den eneste måde at få bugt med en varulv er som bekendt ved at bruge en "silver bullet". Brooks mener imidlertid at "there is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity.". Han fortsætter i samme dur med "we cannot expect ever to see two-fold gains every two years" in software udvikling, ligesom tilfældet har været indenfor hardware udvikling. Dermed konstaterer han at denne såkaldte "silver bullet" ikke eksisterer og ikke vil blive udviklet indenfor de næste 10 år.

I følge Brooks kan grunden til at denne "silver bullet" ikke findes, deles op i to problemer henholdsvis "Essence complexity - Bliver skabt af det problem der skal løses og kan altså ikke fjernes" og "Accidents complexity - Der opstår på grund af fejl udvikleren laver og som dermed kan rettes". Men samtidig hævder han at software udviklerer på dette tidspunkt (1986) bruger det meste af sin tid på "Essence complexity", og mener altså at selvom "Accidental activities" elimineres vil det ikke resultere i en "magnitude improvement" (Tifoldig forbedring i udviklings hastigheden).

Selvom der bliver heftigt argumenteret for at der ikke findes en "silver bullet" mener Brooks alligevel at der er et par lyspunkter på vej, bl.a. i form af høj niveau programmerings sprog som Fortran, der kan sammenlignes lidt med nutidens C og Java.

Brooks proklamerer yderligere at der er forskel på gode system udviklere og fantastiske, idet at han mener software udvikling er en kreativ process og at nogle udviklerer er grundlæggende bedre end andre, grundet deres genetiske arv. At udnytte dette og sørge for at skabe fantastiske udviklere fremfor middelmådige skulle ifølge Brooks kunne lede til en "magnitude improvement".

## Artiklen set i forhold til object orienteret programmering

Brooks nævner blandt andet at han er stor fortaler for object orienteret programmering og ser store muligheder i udviklingen af dette. Han mener at OOP, sammen med et høj niveau programmerings sprog, kunne resulterer i at mindske både "accidental complexity" og "essential complexity" og dermed være en mulig "silver bullet".

Vi (gruppen), mener imidlertid ikke at dette er tilfældet da kompleksiteten ikke kan simplificeres ved hjælp af disse værktøjer, men blot bedre kontrolleres. Vi ser derfor OOP som en teknologi, et koncept altså en måde at angribe et givent problem på. Konceptet tilbyder værktøjer der kan forbedrer et design og derigennem gøre det nemmere at håndterer kompleksiteten, men udvikleren kan ligeså godt ende med et dårligt design der forværrer problemet og dermed kompleksiteten. Så kort sagt, hvis ikke konceptet bruges "rigtigt", risikerer du blot at yderligere besværliggøre opgaven. Vi konstaterer derfor at OOP ikke er en løsning til kompleksitet, men et (af flere) redskaber til at kontrollerer det, hvis udvikleren evner dette.