

# ANMÄLNINGAR

## Hur programmerare arbetar

Under de senaste decennierna har datorprogrammering och mjukvaruutveckling blivit viktiga praktiker i flertalet samhällen och länder. Genomslaget skiftar och utbredningen är fortfarande ojämn, men digitala infrastrukturer och det som krävs för att utveckla och underhålla dessa ser ut att bli alltmer viktigt. I detta framväxande landskap av teknisk och social förändring har nya organisationer och aktörer etablerat sig och fått centrala roller. Företag som Google, Facebook och Apple är svåra att förbise, och deras verksamheter handlar i hög grad om programmering och mjukvaruutveckling. Men framtagandet av digitala produkter sker också i alltifrån mindre spelföretag till företag som arbetar med jordbruk eller flygplansutveckling. Just mjukvaruutvecklingen i dessa senare branscher står i fokus för Espen Suensons avhandling *How Computer Programmers Work – Understanding Software Development in Practice*<sup>1</sup>. Hur kan vi bättre förstå vad som är programmering i till exempel ett företag som jobbar med spel för bärbara digitala prylar som mobiltelefoner och hur arbetar människor med mjukvaruutveckling i branscher där säkerhet är en central faktor?

Avhandlingen är en kombination av ämnena datavetenskap eller programvaruutveckling (Software Engineering) och etnologi, vilket gör den till en avhandling med hybridkaraktär som har varit ovanlig i akademiska sammanhang. I regel så skrivs och bedöms avhandlingar inom ramarna för specifika akademiska ämnen och discipliner. I detta fall har avhandlingen ett

bredare flervetenskapligt anslag, som har skapat vissa möjligheter men också utmaningar.

Forskare från flera olika kultur- och samhällsvetenskapliga ämnen har studerat mjukvara under senare år. Angreppssätten har ofta varit tvärvetenskapliga och präglade av en mix av teoretiska och metodologiska traditioner. Föregångare som Marshall McLuhan och senare Friedrich Kittler liksom Donna Haraway har inspirerat till spekulativa och ibland provokativa forskningsbedrifter. Flera studier har kopplats till nya framväxande fält som STS (Science and Technology Studies, olika former för mediestudier liksom mjukvarustudier (Software Studies). Inom det senare fältet har forskare som Matthew Fuller och Lev Manovich varit tongivande. Suenson nämner några av dessa studier. Han tar dock sedan medvetet avstånd från dem och väljer en annan forskningsväg.

Suensons ambition har varit att bredda perspektivet inom disciplinerna etnologi och datavetenskap genom att fokusera på en ganska smal teoretisk del av etnologi som kombineras med traditionell hermeneutisk teori. Detta kopplas samman med en etnografisk undersökning liksom diskussioner kring programvaruutveckling. Det kan vara en utmaning att bedöma detta breda och flervetenskapliga angreppssätt. Det kräver en öppen läsning som har överseende med att Suenson inte analyserar eller problematiserar hur flera sociala och kulturella dimensioner påverkar materialet. Frågor om identitet, genus, maktförhållanden

<sup>1</sup> Espen Suenson 2015. *How Computer Programmers Work – Understanding Software Development in Practice*. Turku: Department of Information Technologies, Åbo Akademi. 287 pp. ISBN: 978-952-12-3225-1 ISSN: 1239-1883.

eller andra sociala dimensioner ställs aldrig. Avhandlingen utgör därför ett arbete som skiljer sig från flertalet samtida etnologiska studier av arbetsplatser, teknik eller professionella praktiker. Suensons avhandling bidrar dock med något annat, nämligen en rad tankeväckande kopplingar mellan discipliner, teorier och metoder som kan leda till öväntat kunskapsutbyte.

Metodologiskt har Suenson valt att inte använda teori i första hand för att utveckla begrepp eller för att utveckla nya teoretisk kunskap i mötet med det empiriska materialet. Istället tillämpar han kulturteori och hermeneutisk teori på det empiriska materialet på ett sätt som på sätt och vis påminner om kompatibilitetstester inom programutvecklingsvärlden. Det innebär en logik där några valda teorier testas för att se om de kan säga oss något om ett specifikt empiriskt material.

Med hjälp av Hans-Georg Gadamer's teorier och filosofiska arbeten om hermeneutik och retorik liksom kulturteori framförallt hämtad från etnologen Thomas Højrup tar sig Suenson an specifika programmeringspraktiker. Genom avhandlingen diskuterar Suenson kunskapsgapet mellan hur mjukvara och dess utveckling beskrivs och analyseras och vad som sker i det faktiska arbetet vid datorerna. Målet med avhandlingen är inte att direkt förändra mjukvaruutvecklingspraktiker, utan istället att hitta bättre sätt att beskriva hur mjukvaruprogrammerare arbetar.

Avhandlingen baseras på en jämförelse mellan två väldigt olika former av programmering, nämligen spelutveckling och så kallad *safety critical programming* som används i utvecklingen av mjukvara i branscher där säkerhetskraven är väldigt höga. Det handlar alltså om utveckling av upplevelser och nöjesprodukter liksom programmering av produkter där felaktigheter och brister i mjukvaran skulle kunna få oerhört allvarliga konsekvenser. Det mindre spelutvecklingsföretaget Tribeflame analyseras, liksom större företag som arbetar med *safety critical programming*.

En av de etnologiska styrkorna i avhandlingen är hur programmeringspraktiker diskuteras i delstudierna, speciellt i delarna om Tribeflame. Kapitel fem om spelprogrammering och Tribeflame är en etnografiskt intressant del av arbetet. Här blir Suensons etnografiska metod konkret, och den fysiska och materiella kontexten för arbetet med mjukvaruutveckling presenteras. Olika aspekter av programmeringspraktiker i detta specifika empiriska fält tas upp och diskuteras. Från ett etnologiskt och etnografiskt perspektiv kunde just kontexterna, situationerna, platserna och människorna inom denna specifika bransch gärna ha blivit ännu mer undersökta. Dessutom skulle denna typen av material från de andra branscherna, som arbetar med *safety critical programming*, ha stärkt avhandlingens etnologiska kvaliteter.

Även om analyserna av Tribeflame erbjuder intressant etnografiskt stoff, så försvinner en del av det detaljerade etnografiska materialet när mer generella antaganden görs om spelprogrammering. Suenson nämner att spelprogrammering har kopplingar till andra kreativa praktiker, men inga fler jämförelser eller mer utvecklade diskussioner förs om detta. Vad innebär det till exempel att ett företag som utvecklar spel måste förhålla sig till vad användare upplever som roligt (fun) eller underhållande och hur påverkar detta branschen och vad som händer i de specifika situationerna i kontoret? Likaså kunde betydelsen av ett centralt begrepp som säkerhet (safety) i relation till de andra delstudierna ha inkluderats i analysen för att bättre förstå just praktikerna i de branscherna. Hur kunde vi därigenom förstå programmering utifrån de väldigt olika kontexterna som undersöks i avhandlingen? Nu är dessa begrepp (fun och safety) sekundära i Suensons analys, istället läggs fokus på att programmeringen ska leda till användbara (useful) produkter (s. 203). Utifrån ett programutvecklingsperspektiv är detta förmodligen mest relevant, men från en etnologisk horisont vore det fruktbart att sikta på att öka förståelsen för hur programmeringspraktiker är sammanflätade med specifika kulturella kontexter.

Eftersom digital kultur och praktiker som programvaruutveckling och programmering sprids i mer eller mindre global skala är det viktigt med studier som testar nya kombinationer och provar nya vägar till kunskap. Suenson har provat en av dessa vägar. Avhandlingen bidrar till kunskapen om programvaruutveckling på ett intressant sätt. Det finns dock fortfarande en mängd vägar att välja för forskare

som vill lära sig om betydelsen av mjukvara i dagens och framtidens samhällen. Vi behöver fler studier som kan förena undersökningar av ny teknik med kulturella perspektiv och som kan utforska och analysera hur teknik som mjukvara designas, utvecklas, används, föreställs och förändras.

*Robert Willim*

---