**“User-Centered Process for Designing and Implementing Interactive Motion Tracking Software”**

**Introduction**

**Purpose**

**Methodology**

**Tidigare forskning**

Teorin behandlar främst frågor som vad det är som kännetecknar en ACSD och varför en sådan har valt. Dessutom diskuterar den de verktyg som finns för att utvärdera ett system och vilka mått det finns på användarvänlighet.

Ett sådant mått som kan användas i en utvärdering som är passande för ett projekt av denna typ är som jag redan nämnt *”usability”*. Vad som avses med det är helt enkelt att programmet ska kräva så lite arbete som möjligt av användaren för att utföra de uppgifter som systemet är tänkt att kunna göra. Det ska förse användaren med korrekta funktioner med nödvändig information och att de är organiserade på ett naturligt sätt i det grafiska gränssnittet. Att mjukvaran är lätt att lära sig och att saker och ting finns där de förväntas finnas är också ett kännetecken för *”usability”.[[1]](#footnote-1)*

Fokus på ”usability” tenderar ofta att försvinna i delar av mjukvaruprocessen av främst två olika anledningar. För det första handlar det ofta om relationen till kunden där kunden sällan specifikt poängterar vikten av det och när företaget sen levererar produkten till kunden får den en mindre användarvänlig produkt. Här behövs frågan diskuteras tydligare då kunden kanske inte ens känner till konceptet ”usability” även om det på abstrakt nivå är något den önskar. Ett annat problem som kan uppstå är när ”usability”-experter bara är delaktiga i delar av systemutvecklingsprocessen så kommer ”usability” bli lidande i de faser där de inte är delaktiga (vanligtvis implementerings och testningsfasen). Anledningen att de inte är delaktiga är att MDI-experter sällan också är programmerare och därmed är det rena datavetare och systemutvecklare som oftast genomför denna del av utvecklingsfasen. Det som behövs är HCI-experter som har tillräckligt god kunskap om systemutveckling för att kunna delta i implementeringsfasen och ge sin input där. [[2]](#footnote-2)

Det finns två grupper av utvärderingsmetoder, dels har vi expert-utvärderingar där MDI-experter använder olika tekniker för att utvärdera ett givet system eller design. Den andra är användarinvolverade utvärderingsmetoder. Dessa låter antingen de faktiska slutanvändarna, eller om det inte är möjligt någon som kan representera dessa, testa prototypen eller det färdiga programmet. Det finns sedan många metoder för att få ut ett resultat från denna utvärderingsmetod, man kan observera användarna under testets gång eller låta de fylla i formulär eller svara på frågor via intervju. Både expertutvärderingar och användarinvolverade utvärderingar har för- och nackdelar men har man tillgång till slutanvändarna finns de få anledningar att inte låta de testa systemet. Expertutvärderingar ska heller inte göras av den egna designen då den djupgående kunskapen om systemet gör att bedömningen riskerar bli snedvriden. Antingen kan problem som en utomstående expert hade hittat inte dyka upp i utvärderingen eller så hittas obskyra problem som realistiskt i princip aldrig kommer dyka upp i faktiskt användande.[[3]](#footnote-3)

En typ av användarinvolverad utvärdering är kontrollerade experiment vilket är en mer strikt form av prototyptestning där vad som ska utvärderas är tydligt definierat och kvantifierat. Det kan vara jämförelser av två olika gränssnitt där man vill hitta det mest användarvänliga.

För det sistnämnda är *”design principles”* viktiga då det kan underlätta inlärningsfasen genom att applicera vanliga *”design principles”.* Vad som avses då är artefakter som ofta hittas i liknande system, t.ex. ångra/gör om-knappar, att alltid visa för användaren vad som pågår, placera knappar och information på ställen där de hör hemma. [[4]](#footnote-4)

I metoddelen beskrevs redan delar av vad en agil ACSD innebär. Mer i detalj så består hela processen av en fas 0 där grupper av användare identifieras och studeras. Vad är det slutanvändarna kommer att göra med programmet och vad vill de få ut? Detta görs genom ett antal metoder, bland annat definierar man så kallade ”personas” vilket då är en generell beskrivning av en subgrupp av slutanvändarna. Denna beskrivning kan sen användas för att designa för specifika önskemål eller problem som skulle kunna uppstå. Man tar fram ”user stories” och scenarion för att visa på typiska användningssituationer. Allt detta görs genom olika typer av domänmodeller, sekvensmodeller beskriver till exempel olika arbetssekvenser för olika funktionaliteter i programmet. Hur hanteras det och vad är det som krävs av användaren för att genomföra en given uppgift.[[5]](#footnote-5)

Pappersprototyper är också ett bra sätt att tidigt i utvecklingsfasen designa ett optimalt grafiskt gränssnitt genom att enligt användarnas önskemål skapa en prototyp för hur gränssnittet ska se ut och sen låta användarna prova på det. Det går snabbt att modifiera en sådan och användarna kan testa sig fram till vad de föredrar.[[6]](#footnote-6)

1. Benyon, David, *”Designing Interactive Systems”,* s. 84 [↑](#footnote-ref-1)
2. Göransson, B, Gulliksen, J, Boivie, I, *”The Usability Design Process – Integrating User-Centered Systems Design in the Software Development Process”* [↑](#footnote-ref-2)
3. Benyon, D, *”Designing Interactive Systems”, s. 228-235* [↑](#footnote-ref-3)
4. Ibid, s.89 [↑](#footnote-ref-4)
5. Beyer, Hugh, *User-Centered Agile Methods* [↑](#footnote-ref-5)
6. ibid [↑](#footnote-ref-6)