**Preliminär projektplan**

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

**Inledning:**

Examensarbetet kommer genomföras på institutionen för neurovetenskap på Uppsala Universitet där jag kommer arbeta tillsammans med en forskargrupp som studerar rörelsesyn hos blomflugor. Att man forskar på just flugor beror på att det finns tydliga likheter mellan hur rörelsesyn processeras hos flugor som hos människor. För syftet behöver de i detalj kunna studera flugors rörelsebanor och sedermera återskapa dem för analys. Till detta behöver ny mjukvara utvecklas då det är ett specifikt behov som inte uppfylls av redan existerande program. Programmet som behöver utvecklas ska användas av forskarna själva under en längre period. Det ska vara användarvänligt samt fungera enligt dess specifikationer.

Systemet ska läsa input från en spårningsboll som en fastlåst fluga sätter i rörelse genom att röra på benen som om den skulle gå längs en yta. Bollens rörelse läses av med hjälp av optiska sensorer som har tagits från vanliga optiska möss. Dessa läser av rörelsen i 2 dimensioner och systemet ska sedan rita upp rörelsebanan på en 2D-karta. Flugan är också omgiven av en sfär, på insidan av denna kommer datagenererade bilder projiceras med syftet att studera hur synintryck påverkar flugans rörelsemönster.

**Syfte:**

Syftet med detta examensarbete är att utvärdera resultatet av en användarcentrerad systemdesign (ACSD) i praktiken samt producera en användarvänlig och över tid stabil mjukvara för forskning kring flugors rörelsemönster.

**Metod:**

För att uppnå syftet kommer en ACSD genomföras från grunden och appliceras på problemet som ställdes upp i inledningen. Det viktiga är att utvecklingsprocessen faktiskt följer de riktlinjer som kännetecknar en agil ACSD-process samt att den slutliga utvärderingen är väl förankrad i teorin för att konkreta slutsatser ska kunna dra i analysen.

Arbetet kommer delas upp i en inledande fas, en implementeringsperiod som är uppdelad i ett antal sprints, en utvärdering av mjukvaran samt slutligen en fas där resultaten analyseras och slutsatser dras. [[1]](#footnote-1)

Den inledande fasen kommer främst handla om att definiera projektet mer detaljerat än det är nu, hitta och analysera requirements genom att använda de tekniker som en ACSD erbjuder. Bland annat skapa ”user stories”, ”personas” och modellera arbetsmiljön.[[2]](#footnote-2) När detta är gjort kommer en grov design av programmet göras, requirements prioriteras tillsammans med slutanvändarna.

Implementeringsfasen kommer delas upp i sprints vilket kännetecknar en agil utvecklingsprocess. Mål för sprintarna definieras i den inledande fasen när det är tydligare hur den slutgiltiga produkten är tänkt att se ut. Implementeringsfasen är tänkt att pågå i nio veckor så tre sprints om tre veckor är den preliminära uppdelningen. Detaljer för uppdelningen kommer tas fram i den inledande fasen.

Utvärderingen kommer genomföras som en användarinvolverad utvärdering med naiva användare från institutionen. Att naiva användare väljs är av den anledning att det färdiga systemet kan komma att användas av andra universitet som sysslar med liknande forskning. Utvärderingen i sig kommer göras enligt tidigare forskning.

Slutligen kommer resultaten att sammanställas och analyseras för att svara på de huvudsakliga frågeställningarna. I värderingen av resultatet kommer fokus ligga på dels vad som vi kommer fram till i fastställande av requirements men framförallt ”usability”, ”responsiveness”, ”accessability”. Resultatet kommer sättas i ett teoretiskt perspektiv där den tidigare forskning som tas upp i uppsatsen kommer användas.

**Teori:**

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”*. ”Usability” innebär 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 dessa ska vara 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 kännetecken för *”usability”.[[3]](#footnote-3)*

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. [[4]](#footnote-4)

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.[[5]](#footnote-5)

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. [[6]](#footnote-6)

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.[[7]](#footnote-7)

Pappersprototyper är också bra för att tidigt i utvecklingsfasen designa ett optimalt grafiskt gränssnitt. Genom användarnas önskemål skapar man en prototyp för hur gränssnittet ska se ut och sedan låter man användarna prova på det. Det går snabbt att modifiera en sådan och användarna kan testa sig fram till vad de föredrar.[[8]](#footnote-8)

Beskrivningen av de ovanstående verktygen är på en ganska hög nivå och i praktiken blir dessa specificerade enligt hur det enskilda projektet ser ut.

**Vad ska examensarbetet leda till:**

Examensarbetet ska generera system som ska vara ett hjälpmedel i forskningen kring flugors rörelsesyn. Eftersom systemet tagits fram via en ACSD för att skräddarsy det för slutanvändarna ska arbetet också innehålla en utvärdering samt analys av resultatet.

**Tidsplanering**

Vecka 1-3: Inledande fas – Samla och analysera requirements, övergripande programdesign, förbereda arbetsmiljön (sätta upp git repository, inläsning av material) och skriva större delen av syfte, metod och teoridelen av uppsatsen.

Vecka 4-12: Fortsatt design, implementering och testning. Detta görs med fördel i iterationer om 2-3 veckor där milstolpar sätts upp i den inledande fasen.

Vecka 13-14: Utvärdera mjukvara genom att naiva användare testar systemet.

Vecka 15-19: Analysera resultaten från utvärderingen samt färdigställa uppsatsen och förbereda presentation

Vecka 20: Reserv. Alternativt fixa större problem som hittas under utvärderingsfasen.

**Ursprunglig projektbeskrivning**

Master thesis project

Are you interested in developing software that will be used in neuroscience research?  
Do you like programming in Matlab?  
Are you hard working and keen to learn more about motion vision (skills that are highly applicable in the gaming industry)?  
If so, you might consider joining the Motion Vision lab at the Department of Neuroscience (BMC) for a 30 hp project. We are looking for a student who is interested in joining a highly cross-disciplinary research lab at the interface between biology and computer science. The project involves optimization and implementation of target tracking software with high precision requirements. The software will be used to track flies moving against a bright or a cluttered background. The main aim is to generate software that can reconstruct tracks in 2D, but an ambitious student will also have the option to optimize the software for 3D reconstruction, or real-time tracking, which may be implemented with the visual stimulus software. If you have an interest in generating moving stimuli the project could focus on this aspect instead. Training in biology is an advantage, but definitely not a requirement. The lab is English speaking.

1. Beyer, Hugh, *User-Centered Agile Methods* [↑](#footnote-ref-1)
2. Ibid [↑](#footnote-ref-2)
3. Benyon, David, *”Designing Interactive Systems”,* s. 84 [↑](#footnote-ref-3)
4. Göransson, B, Gulliksen, J, Boivie, I, *”The Usability Design Process – Integrating User-Centered Systems Design in the Software Development Process”* [↑](#footnote-ref-4)
5. Benyon, D, *”Designing Interactive Systems”, s. 228-235* [↑](#footnote-ref-5)
6. Ibid, s.89 [↑](#footnote-ref-6)
7. Beyer, Hugh, *User-Centered Agile Methods* [↑](#footnote-ref-7)
8. ibid [↑](#footnote-ref-8)