

Funka Nu.

Tillgängliga pdf-lösningar med
Adobe LiveCycle Designer

Referens:

Andreas Cederbom
andreas.cederbom@funkanu.se
08-555 770 64

Bakgrund

Funka Nu AB har genomfört en undersökning av hur Adobe LiveCycle Designer kan användas för att öka tillgängligheten i pdf-baserade tjänster. Syftet har varit att kartlägga hur lösningar producerade via LiveCycle Designer fungerar tillgänglighetsmässigt, om det finns problem och hur dessa kan hanteras för att skapa lösningar med så hög tillgänglighet som möjligt. Detta har gjorts genom att följa ett projekt där InfoDesign AB jobbat med att ta fram lösningar i LiveCycle Designer för Försäkringskassan. I projektet har Funka deltagit i delprocesser och gett feedback gällande tillgängligheten. Vi har även kompletterat med egna tester i Adobe Acrobat och i LiveCycle Designer.

Funkas metodik är utvecklad i nära samarbete med handikapprörelsen och allt vi rekommenderar är testat i verkligheten. Vår verksamhet bygger på de internationella riktlinjerna för tillgänglighet, Web Content Accessibility Guidelines 2.0 (WCAG 2.0). Funkas långa erfarenhet av tillgänglighetsarbete och tester med användare med olika behov och förutsättningar, med och utan hjälpmedel, visar att WCAG 2.0 inte räcker. Vi har därför själva utarbetat testkriterier för punkter som kompletterar det internationella regelverk som offentlig sektor genom EU-beslut ska följa.

2009 fick Funka i uppdrag av W3C att genomföra en auktoriserad översättning av WCAG 2.0 till svenska. Denna översättning kommer att ligga till grund för det arbete med vidareutveckling av Vägledningen 24-timmarswebben som e-delegationen ansvarar för.

- [Web Content Accessibility Guidelines 2.0 \(WCAG 2.0\)](#)
- [Den auktoriserade svenska översättningen av WCAG 2.0](#)
- [World Wide Web Consortium \(W3C\)](#)
- [Web Accessibility Initiative \(WAI\)](#)

Läs mer om Funka under rubriken ”[Funka Nu AB](#)” i slutet av detta dokument.

Sammanfattning

Syftet har varit att titta på hur Adobe LiveCycle Designer fungerar ur ett tillgänglighetsperspektiv för att skapa e-tjänster. En bra e-tjänst kan guida användaren rätt, hjälpa användaren att undvika samt korrigera fel, och spara tid genom att hämta in information som redan är känd.

För att kunna skapa riktigt bra e-tjänster som guidar användaren på ett bra sätt räcker det inte att jobba med formulär i Adobe Acrobat. Det krävs ett mer intelligent system och då kan LiveCycle Designer vara ett alternativ. Det går att bygga lika bra lösningar i html, men fokus i det här projektet har varit på pdf-gränssnitt skapade via LiveCycle Designer. Vi har inte heller tittat på kombinationen Flash Builder och LiveCycle Designer som gör det möjligt att producera flashbaserade gränssnitt som kommunicerar med LiveCycle Designer.

Lösningar skapade med LiveCycle Designer kan göras mer tillgängliga för användarna än dokument skapade med Adobe Acrobat. Det som går att göra tillgängligt i pdf-formulär skapade via Adobe Acrobat går också att göra med LiveCycle Designer. Fördelen med LiveCycle Designer är att det går att göra mer bland annat med dynamiska formulär.

Med dynamiska formulär kan vi skapa guidade e-tjänster som hjälper användaren på ett bra sätt och gå från vanliga blanketter till riktiga e-tjänster. Tyvärr finns det idag en viss eftersläpning där enstaka hjälpmedel inte hanterar pdf-filer som är sparade som dynamiska filer, men flertalet moderna hjälpmedel stöder det och problemet ligger inte själva formatet.

Det finns mycket att ta hänsyn till för att e-tjänster ska bli tillgängliga oavsett format och program. De stora tillgänglighetsproblemen med pdf skapas inte av LiveCycle Designer eller Adobe Acrobat utan av att lösningar tas fram utan fokus på att göra dem tillgängliga. Det går att skapa tillgängliga lösningar med LiveCycle Designer, men det kräver en hel del av utvecklaren för att säkra tillgängligheten i mer komplexa lösningar.

Vi har hittat en del svårigheter i tillgänglighetsarbetet med LiveCycle Designer. Det är inga avgörande problem men begränsningar som kan vara besvärliga för utvecklarna (se mer i rapporten nedan). Här väntar vi på att Adobe ska utreda var problemen uppstår och på riktlinjer för hur de ska undvikas, alternativt på att problemen ska korrigeras.

Funka Nu AB, Stockholm 2011-03-04.

Innehåll

Adobe LiveCycle Designer	5
Om projektet	5
Tillgängligheten i pdf	5
Vad vill vi uppnå med e-tjänster	5
Tillgängliga tjänster i pdf-format	6
Möjligt att hantera både med mus och tangentbord	6
Tjänster skapade med LiveCycle Designer	7
Läsbart för hjälpmedel	7
Tjänster skapade med LiveCycle Designer	7
Logiskt och begripligt	8
Tjänster skapade med LiveCycle Designer	8
Hjälp med att undvika och korrigera felaktigheter	8
Tjänster skapade med LiveCycle Designer	9
Dynamiska formulär med LiveCycle Designer	9
Taggstruktur för hjälpmedlen	9
Testa lösningen	10
Kontroll av tillgängligheten	10
Stödet för dynamiska formulär i hjälpmedel	11
Resultat	11
Jaws (www.freedomscientific.com): Fungerar	11
SuperNova / Hal (www.yourdolphin.com): Går ej	11
Window Eyes (www.gwmicro.com) Begränsat.....	12
NVDA (www.nvda-project.org) Delvis.....	12
Användarnas olika förutsättningar och behov	13
Funka Nu AB	16

Adobe LiveCycle Designer

Adobe LiveCycle Designer följer med Adobe Acrobat Professional och kan användas för att skapa intelligenta formulär och dokument som inte bara fungerar som statiska sidor utan som riktiga tjänster vilka kommunicerar med bakomliggande system för att skräddarsy information och flöden efter användarens val och önskemål.

Lösningar som utvecklas i LiveCycle Designer resulterar ofta i att användaren får en pdf-fil men man kan också generera html-gränssnitt eller gränssnitt i Flash.

Undersökningen har fokuserat på kombinationen LiveCycle Designer och pdf eftersom det ännu så länge är den vanligaste kombinationen i Sverige.

Om projektet

Projektet har genomförts tillsammans med Försäkringskassan och InfoDesign AB. Funka har varit med i processen när Försäkringskassan tagit fram tjänster i LiveCycle Designer tillsammans med InfoDesign. Funka har gett input i form av tillgänglighetskompetens, testning och rådgivning. I skrivande stund pågår fortfarande Försäkringskassans arbete så vi kan inte säga hur resultatet blir. Våra erfarenheter från det arbetet, tillsammans med egna tester, har lett fram till resultatet i denna rapport.

Tillgängligheten i pdf

Funka gjorde 2005 på uppdrag av Adobe en undersökning av tillgängligheten i pdf-formatet och hur olika hjälpmedel hanterade formatet. Undersökningen visade att formatet som sådant är tillgängligt men att det fanns en viss eftersläpning i stödet i vissa av de testade hjälpmedlen.

De internationella riktlinjerna för tillgänglighet, WCAG 2.0, tar inte ställning till vilken teknik som används för att lägga ut information och tjänster så länge tekniken kan användas på ett tillgängligt sätt. Vår utgångspunkt för den här granskningen har varit att titta på tillgängligheten i filer skapade via LiveCycle Designer, både statiska och dynamiska, samt att göra en jämförelse mot Adobe Acrobat.

Vad vill vi uppnå med e-tjänster

LiveCycle Designer används för att skapa interaktiva lösningar där användarna kan utföra olika tjänster. Det enklaste exemplet är vanliga ifyllnadsbara blanketter. Det går också att göra direkt i Adobe Acrobat men LiveCycle Designer erbjuder betydligt större möjligheter att skapa intelligens i tjänsterna.

Internet har de senaste åren blivit allt mer interaktivt. Det går att göra allt mer direkt över internet utan att användaren behöver skicka in papper, ringa eller

besöka företag och myndigheter. Utvecklingen är positiv och leder i slutänden till mindre arbete både för individen och för organisationen.

Begreppet e-tjänst används idag flitigt och kan inbegripa allt från en blankett som måste skrivas ut och skickas in till ett helt webbaserat system för att hantera ärenden. Det vi vill uppnå är det senare. En e-tjänst ska vara en guide som hjälper användaren att utföra en uppgift. E-tjänsten ska inte bara ta emot data utan hjälpa användaren att fylla in rätt. Det är också viktigt att e-tjänsten verkligen integrerar med bakomliggande system så att användarens arbete inte bara resulterar i ett papper som ska hanteras manuellt.

En viktig del i de internationella riktlinjerna för tillgänglighet WCAG 2.0 gäller felhantering. Det står tydligt i WCAG, att webbaserade system ska hjälpa användaren att undvika att göra fel, och att korrigera de fel som eventuellt uppstår. Det går inte att göra i en statisk blankett som ska skrivas ut och skickas in. För att det ska gå att göra på ett bra sätt krävs att e-tjänsten kan

- kommunicera med bakomliggande system
- hämta tidigare ifyllda uppgifter
- kontrollera ifyllda uppgifter
- guida användaren rätt i flödet.

[WCAG 2.0: 3.3 "Help users avoid and correct mistakes"](#).

Tillgängliga tjänster i pdf-format

För att en tjänst ska fungera på ett tillgängligt sätt krävs att utvecklaren/redaktören arbetar efter några viktiga principer. Nedan går vi igenom de viktigaste punkterna och redogör för hur LiveCycle Designer fungerar samt vilka tillgänglighetsproblem vi sett under projektets gång. Att vi skriver om ett problem betyder inte att det inte går att lösa, utan att vi inte sett någon lösning under det här projektets gång.

Möjligt att hantera både med mus och tangentbord

Det måste gå att fylla i och hantera hela tjänsten/formuläret enbart med mus och enbart med tangentbord oberoende av vartannat. Detta gäller oavsett om användaren har ett hjälpmedel eller ej. Användaren ska inte behöva använda både mus och tangentbord för att kommunicera med e-tjänsten. Det fungerar oftast på ett bra sätt utan att man som utvecklare eller redaktör behöver tänka på det, men ju mer interaktivitet som byggs in desto viktigare är det att:

- **Kontrollera att det går att tabba sig mellan olika objekt med tangentbordet.** Genom att upprepade gånger trycka på tabb-tangenten (till vänster om Q på svenska tangentbord) förflyttar man fokus på en sida från formulärobject till formulärobject och från länk till länk. Det ska också gå att göra de val som krävs för att utföra tjänsten. Normalt sett fungerar det här bra. Använder man standardkomponenter för textfält, kryssrutor, listor, knappar och så vidare, så blir det bra. Men om man exempelvis börjar arbeta med javabaserade områden där användaren ska peka för att det ska

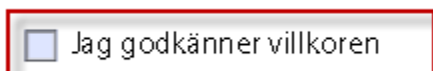
komma fram hjälptexter kan man skapa stora problem för användare som navigerar med tangentbordet.

- **Kontrollera tabbordningen.** Användaren ska inte bara kunna ta sig mellan de olika objekten och fylla i information via tangentbordet, det måste också gå att komma åt de olika delarna av tjänsten i en logisk ordning. Tabbordningen måste därför anges i filen.
- **Kontrollera att klickytorna är rimligt stora.** Det ska gå att fälla ut menyer, klicka på ikoner och knappar även med nedsatt precisionsförmåga eller i pekskärmsgränssnitt där fingret som klickar kan täcka en ganska stor del av gränssnittet.

Tjänster skapade med LiveCycle Designer

Eftersom LiveCycle Designer erbjuder utvecklaren stora möjligheter att bygga interaktivitet med skript finns det också stora möjligheter att skapa lösningar som inte går att hantera med tangentbordet. Så länge man håller sig till de inbyggda komponenterna, exempelvis textfält, radioknappar, kryssrutor, listor och så vidare, så fungerar det bra. Men om gränssnittet får menyer som är skriptbaserade måste man testa och säkra att dessa även fungerar med tangentbordet på ett enkelt sätt.

När det gäller klickytor finns ett problem med LiveCycle Designer idag. I html kan man, om en kryssruta och ledtext/etikett är rätt inlagd, klicka på hela ledtexten för att fylla i kryssrutan/radioknappen. Men i pdf fungerar det inte på samma sätt, här måste användaren klicka i själva radioknappen eller kryssrutan.



Figur 1: Hela det inramade området bör vara klickbart för att kryssa i kryssrutan, men i pdf är det bara den blå fyrkanten som är det.

Det här problemet finns även i Acrobat, det är alltså inte sämre i LiveCycle Designer utan är ett generellt problem.

Läsbart för hjälpmedel

Det finns många olika hjälpmedel för att hjälpa användaren att läsa text. De skiljer sig också åt beroende på om de riktar sig till användare med läs- och skrivsvårigheter eller användare med synnedsättning. För att dessa typer av hjälpmedel ska klara av att hantera innehållet i en pdf-fil krävs att allt är rätt inlagd.

- **Undvik att använda bilder av text.** En bild som föreställer text kan inte tolkas av hjälpmedel för personer med lässvårigheter eller synnedsättning.
- **Säkerställ att all text går att komma åt med hjälpmedel för gravt synskadade.** En seende användare kan peka på den text som ska läsas upp men hjälpmedel för gravt synskadade fungerar så att användaren måste pila eller tabba sig ner i gränssnittet. Då måste informationen presenteras i rätt ordning och all information måste vara åtkomlig.

Tjänster skapade med LiveCycle Designer

Givetvis går det att skapa stor otillgänglighet med LiveCycle Designer genom att exempelvis lägga in bilder av text i stället för riktig text, men det är ingen skillnad mot andra typer av tekniker och program.

Ett problem för användare som ser dåligt är att man inte uppfattar förklaringar och förtydliganden som ligger i närheten av fältet. Det är exempelvis inte ovanligt att formulär ser ut så här:



Personnummer *
(Exempel: 19700909-0909)

Figur 2: Etiketten till fältet för personnummer anger bara att det är ett personnummer som ska fyllas i men under fältet står det hur personnumret ska vara skrivet.

En gravt synskadad användare förflyttar sig mellan fälten med tabbtangenten och uppfattar därför inte om det finns text mellan dessa. För att hjälpa användaren att uppfatta texten kan förklaringen/exemplet läggas in i ett attribut för fältet. Det finns flera olika att välja på i LiveCycle Designer, men de vanligaste är ”Tool tip” och ”Custom Screen Reader text”. Det går också att ange vilket av dessa som ska läsas av hjälpmedlen.

Tyvärr verkar det just nu inte gå att visa ett attribut visuellt och få det andra att läsas av hjälpmedlet. Idealt ska ”Tool tip” användas för en fördjupande text som visas visuellt när användaren pekar på fältet med musen, medan ”Custom Screen Reader text” används av uppläsande hjälpmedel för gravt synskadade. Om man anger två olika texter för ”Tool tip” och ”Custom Screen Reader text” och anger att det är ”Custom Screen Reader text” som ska läsas av hjälpmedlen tycks det medföra att det även ersätter ”Tool tip” när man pekar på fältet med muspekaren.

Ett annat problem som uppstår i hjälpmedel för gravt synskadade är att skrivskyddade fält har en tendens att upprepas när man navigerar med tangentbordet (de dyker upp som två separata tabbstopp). Det tycks gälla både i dynamiska och statiska formulär, oavsett om de skapas med LiveCycle Designer eller med Adobe Acrobat. Problemet är inte allvarligt men kan skapa en viss förvirring.

Logiskt och begripligt

För alla användare är tjänstens logik och begriplighet avgörande för om man tar sig igenom den eller ger upp. Det finns idag ett ändlöst antal exempel på tjänster som är krångliga, ologiska och besvärliga. Det skapar frustration hos användaren och onödigt mycket byråkrati och support hos organisationen som står bakom tjänsten.

Tjänster skapade med LiveCycle Designer

Det är här som LiveCycle Designer verkligen kan höja tillgänglighetsnivån i jämförelse med vanliga formulär skapade i Adobe Acrobat. Med LiveCycle går det att göra kopplingar mot bakomliggande tjänster och system, hämta data ur databaser och integrera i tjänsten. Det finns exempelvis inga tekniska hinder för att bygga en tjänst där användaren bara behöver mata in sitt personnummer och sedan skriver tjänsten in alla relevanta personuppgifter som behövs.

Det går att skapa guider och tjänster där efterföljande steg anpassas efter de val som matas in i tjänsten. Då visas bara de fält som är aktuella för användaren.

Hjälp med att undvika och korrigera felaktigheter

Som vi nämnt tidigare så innehåller WCAG riktlinjer om att hjälpa användaren att undvika misstag och att korrigera gjorda misstag. För att det ska fungera måste

tjänsten ha byggt in validering av data. Idag är ett vanligt problem att formulär visserligen görs ifyllnadsbara men de saknar ofta en kontroll av de uppgifter som matas in. Det leder till att myndigheten/företaget får in blanketter/handlingar som inte är korrekta eller kompletta. Då måste användaren komplettera och korrigera i efterhand. Det kostar både tid och pengar, både för användaren och för organisationen.

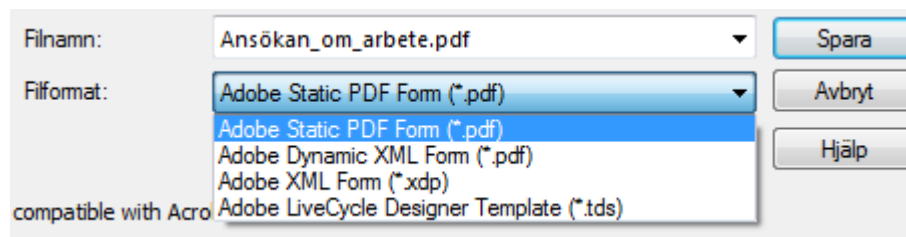
Tjänster skapade med LiveCycle Designer

Det går att bygga in intelligent validering som direkt ger feedback till användaren om denne skrivit in rätt eller fel. Informationen behöver inte gå iväg till en handläggare med en felaktig e-postadress eller om användaren glömt mata in uppgifter som måste finnas med för att behandla ärendet. Genom att lägga in intelligenta kontroller kan tjänsten minska handläggningstid, supporttid och korrespondens för att få in kompletterande uppgifter och rättelser.

Möjligheten att läsa in redan kända data minskar också risken för fel väsentligt.

Dynamiska formulär med LiveCycle Designer

För att LiveCycle Designer ska klara av att dynamiskt ändra vilka fält och texter som ska visas i en pdf-fil så krävs det att strukturen i filen kan ändras efterhand. Därför går det också att spara filer i två olika pdf-format, antingen som statisk pdf eller dynamisk pdf. Som statisk pdf är strukturen helt klar i förväg och det begränsar möjligheterna till interaktivitet i tjänsten.



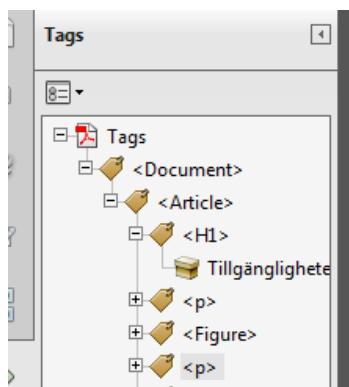
Figur 3: Det går att spara som statisk eller dynamisk pdf, eller xml.

För att kunna dra nytta av LiveCycle Designers möjligheter måste vi alltså spara som dynamiskt formulär i de flesta fall.

Taggstruktur för hjälpmedlen

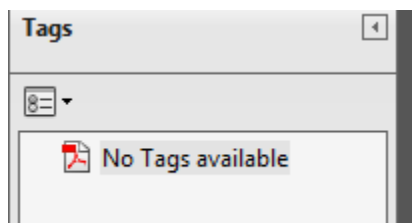
För att framförallt hjälpmedel för gravt synskadade ska kunna läsa och återge en pdf-fil på ett bra sätt använder de sig av något som kallas Taggar/Tags. Dessa bildar en dold struktur i pdf-filen som anger ordningen på olika objekt och också vad de är. Taggstrukturen anger exempelvis vad som är rubriker, hur tabeller är organiserade och vad som är vanlig text. Det absolut vanligaste tillgänglighetsproblemet i pdf-filer idag är att de skapats utan dessa taggar av programvaror som inte klarar av att hantera det eller av redaktörer/utvecklare som inte känner till hur man gör.

Det går att inspektera taggstruktur i Adobe Acrobat Professional genom att visa taggar. Det kan se ut så här:



Figur 4: Taggar som beskriver ordning och vad innehållet är. H1 står för huvudrubrik, P för stycke och Figure för en bild.

Problemet är att denna struktur måste förändras i en dynamisk pdf. Eftersom innehållet ändras beroende på valen användaren gör kan inte strukturen vara fixerad i förväg. Om man därför letar efter taggarna i en dynamisk pdf kommer man inte att hitta dem.



Figur 5: Så här ser det ut om du visar taggar i en dynamisk pdf-fil.

Taggarna i en dynamisk pdf-fil bildas i stället efterhand för hjälpmedlen. Filen kan alltså se otillgänglig ut (den saknar taggar), men fungera fullt tillgängligt.

Testa lösningen

Utvecklingen mot allt mer komplexa tjänster gör att det blir svårare att förutsäga hur olika typer av hjälpmedel och läsare ska bete sig. Utvecklar vi tjänster med interaktiva, dynamiska formulär, så måste de testas med riktiga hjälpmedel för att säkerställa att de fungerar på ett korrekt sätt.

Problem kan uppstå på många olika nivåer, de kan uppstå i programvaran som används för att skapa gränssnittet, de kan uppstå i en läsare (exempelvis Internet Explorer eller Adobe Reader), de kan uppstå i hjälpmedlen eller så uppstår de hos användaren eftersom genomförandet med hjälpmedlet i den aktuella lösningen blir för svårt.

Genom att testa kan vi identifiera eventuella problem och vidta åtgärder för att minska eller eliminera problemen.

Kontroll av tillgängligheten

För att kontrollera tillgängligheten i en dynamisk pdf-fil måste man testa med riktiga hjälpmedel. Det går inte att enbart utifrån källfilerna kontrollera hur det kommer att fungera. Ju komplexare gränssnitt som byggs desto mer tester behövs.

Utöver att testa med hjälpmedel bör man också testa de punkter vi tagit upp ovan, exempelvis att klickytorna är tillräckligt stora, att det fungerar att hantera gränssnittet både med tangentbordet och med mus samt att lösningen är logisk och begriplig.

Stödet för dynamiska formulär i hjälpmedel

Under projektet har vi testat ett antal hjälpmedel tillsammans med formulär skapade med LiveCycle Designer. Fokus har legat på hjälpmedel för gravt synskadade eftersom det är dessa som i första hand har nytta av den taggstruktur som måste skapas ”on demand” i dynamiska formulär. Det är alltså där tillgänglighetsproblemen troligast uppstår i kommunikationen mellan hjälpmedel och gränssnitt.

Statiska formulär fungerar på samma sätt oavsett om de skapats i LiveCycle Designer eller direkt i Adobe Acrobat. Det har ingått som en del av testerna man vi har inte sett några nämnvärda skillnader som beror på var lösningen skapats varför vi inte redovisar testerna i detalj här. De tillgänglighetsåtgärder man kan göra i Adobe Acrobat kan man också göra i LiveCycle Designer. Därför har fokus legat på dynamiska lösningar. Samtliga hjälpmedel hanterar statiska formulär på ett relativt bra sätt.

Det är stor skillnad på hur dynamiska formulär fungerar med hjälpmedlen. Det råder en viss eftersläpning hos några tillverkare, precis som var fallet med vanliga pdf-filer år 2005, vid den förra undersökningen. Vi har testat de vanligaste skärmläsarna på svenska marknaden.

Vi har varit i kontakt med leverantörerna av hjälpmedel i frågan om stödet för tillgänglighet. Bland annat hos Dolphin jobbar man nu på att bygga in stödet men i skrivande stund är det oklart när det kan komma.

Resultat

Jaws (www.freedomscientific.com): **Fungerar**

Vi har här testat med Jaws 9 (svensk), Jaws 10 (svensk), Jaws 11 (svensk) och Jaws 12 (engelsk, den i skrivande stund senaste versionen).

Detta är den största skärmläsaren på den svenska marknaden och också en av de ledande internationellt sett. Stödet för pdf har funnits länge och det går även bra att hantera dynamiska filer med Jaws version 9 och framåt. Äldre versioner har inte primärt problem med dynamiska pdf-filer utan snarare med nyare versioner av Adobe Reader.

SuperNova / Hal (www.yourdolphin.com): **Går ej**

Vi har enbart testat med SuperNova 12 (svensk). Eftersom denna inte haft något stöd för dynamiska pdf-filer är det inte relevant att testa äldre versioner.

SuperNova hette tidigare Hal. Detta är den näst största skärmläsaren på den svenska marknaden. Här har stödet för pdf varit sämre än hos Jaws. SuperNova 12 (den senaste versionen) erbjuder ett bra stöd för statiska pdf-filer men hanterar tyvärr inte dynamiska.

Window Eyes (www.gwmicro.com) Begränsat

Vi har testat med Window Eyes 7.2. Detta är den för närvarande senaste versionen. Eftersom detta är en mindre skärmläsare på den svenska marknaden har vi valt att inte testa äldre versioner.

Stödet för pdf har funnits länge och är bra om än inte lika bra som hos Jaws. För närvarande är stödet för dynamiska pdf:er begränsat. Det går bra att läsa formulärfältens ledtexter/beskrivningar men det är komplicerat att fylla i dem. Detta beror på att Window-Eyes läser in sidan på nytt efter varje inskrivet tecken det innebär att skärmläsarens formulärläge slås av och måste aktiveras igen för att det skall vara möjligt att skriva vidare. Det är möjligt för en avancerad användare att komma runt dessa svårigheter men det kräver kunskaper som en normalanvändare inte behärskar.

NVDA (www.nvda-project.org) Delvis

Vi har testat med NVDA 2010.2. Detta är den för närvarande senaste versionen. Vi har inte bedömt att det varit relevant att testa äldre versioner. Eftersom det är en gratis skärmläsare som användarna i Sverige själva får söka upp och installera finns det inga hinder för att använda den senaste versionen i dagsläget.

NVDA har ett bra stöd för pdf. I stort sett fungerar NVDA likvärdigt med Jaws. Det enda problem vi noterat är att det inte går att använda piltangenterna för att förflytta sig i dynamiska pdf:er. Detta medför att användare riskerar att gå miste om text som placerats mellan de olika formulärfälten eftersom sådan text inte kan nås med hjälp av tabbtangenten. Den har även problem att läsa beskrivningar på knappar.

Användarnas olika förutsättningar och behov

Alla människor är olika. Vi har olika bakgrund, egenskaper och behov. Olikheterna beror dels på våra fysiska förutsättningar och dels på situationen. Detta ställer stora krav på utformningen av olika typer av gränssnitt, exempelvis bankomater, biljettautomater och webbaserade gränssnitt. Sitter du vid din vanliga dator har du förhoppningsvis ställt in denna så att du har goda möjligheter att se, läsa och ta till dig informationen. Är du ute och reser och surfar med din mobiltelefon har du helt andra förutsättningar. Många gånger blir det stora problem att se och förstå gränssnittets uppbyggnad och funktion. Genom att bygga tillgängliga webbplatser fungerar de bättre i alla situationer du kan hamna i.

Det finns en rad olika situationer och förutsättningar som en användare kan ha eller hamna i som påverkar förmågan att ta del av gränssnittet. Här är bara några av alla dessa.

Användare med problem att styra datorn

En användare kan ha problem med att styra datorn. Problemet kan bero på ett funktionshinder, exempelvis reumatism, men det kan också bero på utrustningen. Få mobiltelefoner på marknaden ger användaren möjlighet att "klicka". I stället använder man ett sorts tangentbordsliknande navigeringssätt, där man tar sig från länk till länk med tryck på en speciell knapp. Samma knapp finns på ditt vanliga tangentbord och heter där tabb. Webbplatsen ska gå att styra oberoende av om du har en mus att klicka med eller ett tangentbord att navigera med. Det kallas för "Input Device Independence".

Det finns en rad olika styrhjälpmedel till datorn. Exempelvis talstyrning, huvudstyrda möss och fotnavigering. Fungerar gränssnittet "Input Device Independent" så fungerar det även för dessa hjälpmedel.

Många användare har problem med precisionen. Därför är det viktigt att det inte krävs onödigt stor precision för att klicka rätt i gränssnittet. Problemen kan bero på darrhänthet eller att du sitter på ett skumpande tåg. Till denna kategori hör också en stor grupp med personer som har arbetsskador, exempelvis personer med förslitningsskador, belastningsskador eller andra typer av arbetsrelaterade skador.

Användare med problem att tolka och förstå

För att besökaren ska kunna använda gränssnittet krävs att han eller hon förstår vad man ska göra i webbtjänsten. En ovan användare har ofta stora problem med detta eftersom denne kanske inte har sett samma typ av tjänster/lösningar tidigare. Det kan vara svårt att navigera, man glömmer kanske scrollister och har svårt att förstå hierarkier på nätet. Därför är det viktigt att gränssnittet fungerar konsekvent och att länkar och menyer är bra utformade.

Sitter du vid skärmen på en bärbar dator i en mycket ljus miljö är det ofta svårt att uppfatta olika färgskiftningar. Därför är det viktigt att informationen inte är färgkodad så att du måste uppfatta olika nyansskillnader för att göra rätt. Även personer med kognitiva funktionshinder och personer som befinner sig i stressade miljöer har ett stort behov av enkla, tydliga och pedagogiska gränssnitt.

Användare med problem att läsa och tolka skriftlig information

I Sverige finns det många människor som har svårt att läsa och skriva. Förutom en stor grupp dyslektiker finns det många som har problem med läsförståelsen utan att räknas som dyslektiker, exempelvis invandrare, döva och ovana läsare. Cirka en femtedel av Sveriges befolkning har någon form av läsproblem. För att hjälpa dessa användare krävs en ren, pedagogisk och anpassningsbar visning i gränssnittet, förutom att texterna bör struktureras med det viktigaste först, hållas korta samt skrivas med ett enkelt språk.

Dyslektiker kan använda hjälpmedel som läser upp texten, så kallade skärmläsare. Det finns även hjälpmedel som gör det möjligt att ändra visningen av texten eller underlättar läsningen på annat sätt. Men för att hjälpmedlen ska fungera krävs att webbplatsen är tekniskt tillgänglig.

Användare med problem att se

Många svenskar har någon form av synproblem. De flesta kan korrigera synproblemen med glasögon eller linser, men det finns situationer där det ändå inte fungerar. En tillgänglig webbplats ger besökaren möjlighet att ställa in visningen, exempelvis textstorleken. Det är väsentligt för en person med synproblem.

Även om man har perfekt syn kan det ibland vara svårt att se, till exempel om man använder en mindre enhet, exempelvis mobiltelefon eller PDA, sitter i en dålig ljusmiljö eller är trött och stressad.

Personer med kraftigare synskador räknas som funktionshindrade. Dessa användare har många olika typer av hjälpmedel. Till exempel skärmläsare som förmedlar det som visas på skärmen via en talsyntes och/eller en punktskriftsdisplay vilket gör att användaren kan "lyssna" eller "känna" på webbplatsen. Skärmläsare för synskadade skiljer sig från skärmläsare för dyslektiker. En gravt synskadad behöver få struktur, bilder och grafiska objekt presenterade genom skärmläsaren. För att det ska fungera krävs det att webbplatsen är tekniskt tillgängligt konstruerad.

Hur många berörs?

Det är svårt att exakt säga hur många personer som har problem att använda en webbplats på grund av bristande tillgänglighet.

Följande siffror visar dock att det rör sig om ett mycket stort antal potentiella användare med avvikande förmåga:

Dyslektiker	600 000
Synskadade	175 000
- varav blinda	14 000
Dövblinda	1 200
Rörelsehindrade	600 000
Reumatiker	800 000
Svårt tal- och språkskadade	40 000
Personer med Neuropsykiatriska/neuropsykologiska problem (DAMP/ MBD, MPD, ADHD, autism)	120 000
Personer med begåvningsnedsättning	400 000
- varav utvecklingsstörda	40 000
- - varav gravt utvecklingsstörda	16 000
Hörselskadade	780 000
Döva	14 000
- varav födda döva	10 000

Källa: Hjälpmedelsinstitutet

Cirka 1,3 miljoner människor i Sverige har någon form av funktionshinder. Och om vi då vill hitta en siffra på hur många som kan tänkas ha problem att använda Internet tillkommer en rad grupper. Exempelvis människor med svenska som andraspråk, äldre användare, teknikovana användare, och de som har koncentrationssvårigheter. Dessutom bör vi räkna in de personer som har problem i olika situationer, exempelvis för att de för tillfället använder en mindre enhet (exempelvis mobiltelefon) eller sitter i en dålig ljus- eller ljudmiljö. Sammantaget är detta en stor och disparat grupp människor som i dag inte utan besvär kan använda Internet. Man kan fråga sig hur en ”normalanvändare” egentligen ser ut.

Funka Nu AB

Funka startade som ett projekt inom handikapprörelsen. I dag är Funka marknadsledande inom området tillgänglighet med 80 procent av Sveriges myndigheter som kunder. Sedan år 2000 är vi ett privatägt bolag och vår nära relation till handikapprörelsen innebär en unik kvalitetskontroll.

Funkas arbete sätter normer för utveckling och analys samt utformar krav vad gäller tillgänglighet. Vi finns med i de flesta internationella arbetsgrupper av betydelse och genomför regelbundet egna undersökningar. Funka driver därmed tillgänglighetsfrågan aktivt både i Sverige och EU.

Funkas konsulter har varit med och tagit fram Handikappombudsmannens riktlinjer för en tillgänglig statsförvaltning och Vervas Vägledning för 24-timmarswebben. I Vägledningen har vi inarbetat Funkas metodik i avsnittet om tillgänglighet. Funka har dessutom drivit fram rekommendationer kring snabbkommandon och ikoner.

Funka är EPiServer Solution Partner, Microsoft Partner och Adobe Certified Training Provider. Vi står bakom Funkaportalen, en mötesplats för funktionshinderfrågor.

Bolagsfakta

Styrelsens säte: Stockholm

Styrelsen består av: Jan Friedman (ordförande), Mats Wennberg, Lennart Engelhardt och Johan Jakobsson.

Antal anställda 2010: 17

Omsättning 2009: 12 MSEK

www.funkanu.se