

Avskaffa användbarhetsarkitekten

Reflektioner kring en användbarhetsarkitekts vardag 2001

Pelle Mårtenson

Softronic AB

+46 (0)708-81 32 86

pelle.martenson@softronic.se

SAMMANFATTNING

I dag tvingas många användbarhetsarkitekter lägga ner alltför mycket energi på att argumentera för sin egen existens inför kunder och medarbetare. Användbarhet efterfrågas inte i så stor omfattning att det har blivit en naturlig del i varje systemutvecklingsprojekt. Ett alternativ till att försöka sälja in och argumentera för användbarhet är att tvinga in ett användarcentrerat arbetssätt i utvecklingsprocessen och därmed styra tillgången i stället för efterfrågan. På så sätt slutar man att erbjuda oanvändarcentrerad systemutveckling som ett alternativ till kunderna. En strategi för att lyckas med detta är att smyga in ett användarcentrerat tankesätt tidigt i IT-utbildningarna och få utvecklarna att ta större del i användbarhetsarbetet. En annan är att användbarhetsarkitekter tar över arbetsuppgifter som traditionellt har legat hos tekniker. Lyckas det kan dagens användbarhetsarkitekt avskaffas och en ny träda i kraft, en som inte ägnar dyrbar arbetstid åt att försvara sitt existensberättigande, utan designa användbara IT-system tillsammans med användarna, kunden, och utvecklarna. Artikeln är skriven utifrån erfarenheter av användbarhetsarbete i IT-konsultindustrin.

1. LÄGET IDAG

Varje samfund som behöver rättfärdiga sin existens har problem. Så skriver John S. Rhodes på webbplatsen Webword, och kanske är det just det som är själva kärnan i de problem som finns med att göra användbarhet till en självklar del i varje produktutvecklingsprocess.

På andra fronter betraktas utveckling av arbetsmetoder, verktyg och material som någonting positivt. Att i IT-branschen införa en "ny" roll, användbarhetsarkitekten med sina "nya" metoder och arbetssätt har betraktats med skepticism och misstanke av en del kunder och medarbetare. Varför ska vi betala för en sådan? Vad tillför en sådan? Om man klarade sig utan sådana förut varför måste vi ha då ha en nu?

Dessa frågor tror jag säkert att de flesta som jobbar med användbarhet har stött på, både från kunder och medarbetare. En ständigt återkommande del i en användbarhetsarkitekts vardag är att slipa på sina argument för sitt eget existensberättigande. Varför ska vi över huvud taget behöva göra det? Det är dags att sluta slipa på argumenten och i stället göra dem överflödiga.

2. TILLGÅNG OCH EFTERFRÅGAN

Det stora problemet är inte att inlemma användbarhetsarbetet i organisationen. Man behöver inte göra det krångligare än det är. En bra början är att några av projektmedlemmarna ställer och utgår från frågor som: Vad är syftet med systemet? Vilka är användarna? Hur jobbar de? Vad har de för utbildning? Ställs liknande frågor tidigt i projekten kommer man att lyckas betydligt

bättre än i dagens genomsnittliga IT-projekt. Även om förståelsen finns att användbarhet och användarcentrerad systemutveckling är något bra är det långt ifrån ett självklart begrepp inom systemutveckling.

Konsultbranschen styrs av ordet debiteringsgrad, vilket är ett mått på hur stor andel av konsultens arbetstid som någon kund betalar för. De med hög debiteringsgrad är konsultföretagets hjältar. Om kunderna skulle kräva användbarhet i större grad skulle vi säkert också lyckas göra det till en naturlig del av utvecklingsprocessen, det skulle inte finnas något val. Visst vill många ha användbara produkter, men få verkar vara villiga att betala priset. Kvalitet lönar sig ofta i längden men kan svida i plånboken för stunden.

När jag först kom till mitt nuvarande arbete benämndes vi "De nya kompetenserna". Efter ett tag när det visade sig att man inte lyckades så bra med insäljandet av användbarhet myntades ett nytt begrepp, "De svårsålda kompetenserna".

Trots gedigna övertygnings- och argumentationskampanjer från användbarhetsarkitekter världen över verkar det vara en bit kvar till efterfrågad användbarhet. Många gånger får inte ens användbarhetsarkitekten träffa kunden i de lägen där detta insäljningsarbete kan göras. Och även om man får det väljer kunden oftast att tänka kortsiktigt, även om de förstår de långsiktiga användbarhetsargumenten.

Det säger sig självt att om nyttan med användbarhet var en allmän självklarhet skulle vi inte ständigt behöva försvara vår roll utan få en betydligt trevligare vardag. Kan vi inte övertyga våra kunder om det och få dem att kräva användbarhet måste vi göra det så självklart att det inte kommer på fråga att sälja ett IT-projekt utan användbarhet. På så sätt ändrar vi den andra variabeln i ekvationen, tillgången.

Det är inget nytt att styra efterfrågan genom att ändra tillgången. Speciellt inte när det gäller ny teknik. Ett talande exempel som alla känner till är cd-spelaren. Själv var jag sen med att skaffa cd. Jag gillade mina gamla LP-skivor, men till slut ströps tillgången eftersom teknikbranschen inte längre ville stödja den gamla vinyltekniken, och även jag var tvungen att skaffa cd. Just nu

håller samma sak på att hända med VHS-videon kontra DVD-spelaren.

3. GÖR OM IT-UTBILDNINGARNA

En strategi för att ändra tillgången är att vända uppochner på alla IT-utbildningar. I dag börjar man på de flesta högskolor så gott som alltid med att läsa matte och grundläggande programmeringstekniska kurser. Man går från ettor och nollor vidare till programmering, lär sig algoritmer och avancerad teknik och når slutligen, om man har tur upp till ytan till någon enstaka kurs i MDI.

Denna ordning motsvarar hur ett IT-system beskrivs från grunden ur ren teknisk synvinkel och inte det ultimata sättet att bedriva systemutveckling. I dag vet vi att det inte är någon bra idé att börja med tekniken i ett IT-utvecklingsprojekt. Vi vet att man måste börja i andra änden och träffa de personer som ska använda produkten, lära sig vilka de är, analysera deras behov och ta reda på vad syftet med systemet är.

Varför inte smyga in det användarcentrerade arbetssättet redan från början i alla IT-utbildningar? Detta arbetssätt måste dunkas in i ryggraden från början och följa hela utbildningen, istället för att bli en överraskning mot slutet. Inte förrän alla i branschen får in det användarcentrerade tankesättet i ryggraden kan vi känna oss riktigt nöjda.

4. DEN NYA SKOLAN

Om man från början lär sig grundläggande användbarhetskunskaper bör dessa användas i alla kommande programmeringsuppgifter. Varför inte kräva att alla programmeringsuppgifter måste ha en beskrivning av tänkta användare och vilka speciella användbarhetsaspekter som måste beaktas i varje uppgift, innan de påbörjas? I början av utbildningen kan det handla om att ge studenterna några enkla frågor att ta reda på genom användarcentrerade metoder. I de högre årskurserna bör studenterna själva lära sig att ställa rätt frågor.

Låt oss ta ett exempel på en tänkt programmeringsuppgift tidigt i utbildningen, där studenterna ska göra ett mycket enkelt banksystem (detta brukar vara ett populärt

exempel i programmeringslitteratur). Nedanstående frågor skulle då kunna inleda uppgiften:

- Vilka är användarna?
- Vad är syftet med systemet?
- På vilken form ska informationen matas in i systemet? Kr, TKr, €?
- Ska man kanske kunna mata in informationen på valfritt sätt?
- Ska informationen som kommer ut ur systemet presenteras på samma sätt?
- Är det bäst ur användarsynpunkt att mata in löner som års- eller månadslöner?

Dessa frågor skulle kunna användas i en mycket enkel uppgift som inte ens har ett grafiskt gränssnitt. På detta sätt väver man in användbarhetstankegångar på ett naturligt sätt redan från början i utbildningen och mer eller mindre tvingar fram ett användarfokus.

Strategin går alltså ut på att sprida en del av det användarcentrerade tankesättet till de som är med i alla systemutvecklingsprojekt, utveckla, genom att göra detta tankesätt till en självklar del även i deras roll.

5. INFILTRERA SYSTEMDESIGNEN

En annan strategi, för att ändra tillgången, är att som användbarhetsarkitekt i högre grad infiltrera områden som traditionellt har sköts av tekniker. Ett ganska vanligt scenario är att användbarhetsarkitekten kommer in i slutet och testat en design som tekniker har gjort, med små möjligheter till påverkan. Och även om användbarhetsarkitekten lyckas komma in tidigt i projektet och bidrar till grunddesignen genom att ta fram prototypunderlag tappas ofta kontakten med projektet när prototyperna är levererade. Detta medför ofta att designen vattnas ur genom att delar misstolkas och konsekvens ej uppehålls.

Det kan ofta vara tacksamt att ta över ansvaret för designen av gränssnittet och interaktionsprinciperna från utvecklare, som hellre ägnar sina pressade tidsscheman åt programmeringstekniska frågor. Ju mer insyltad och ju större ansvar som läggs på användbarhetsarkitekten ju mer tvingas det användarcentrerade tankesättet in i utvecklingen. Och egentligen handlar ju även form- och användbarhetsarbete om systemdesign.

Fördelen med att jobba med systemdesign i stället för användbarhet är att det är en naturlig och ej ifrågasatt del i utvecklingen, även om det till stor del handlar om rent användbarhetsarbete.

6. ANVÄNDBARHETSARKITECTENS DÖD

Genom dessa båda strategier tvingas ett användarcentrerat tankesätt in i utvecklingen och användbarhetsarbetet kommer att finnas med i IT-projekt utan att behöva säljas in och argumenteras för. Vi har ändrat tillgången i stället för efterfrågan. Kunden behöver då inte ens fundera på frågan: Användbarhet eller inte? Denne kommer inte att ha något val. På samma sätt som kunden i dag knappast kan välja att få sitt IT-system utvecklat i Cobol. "Det är ett gammalt verktyg. Vi är ett modernt företag och använder inte omoderna verktyg. Vi använder inte heller omoderna metoder, såsom oanvändarcentrerad systemutveckling."

Kan vi inte öka efterfrågan får vi helt enkelt ändra tillgången. "Oanvändarcentrerad systemutveckling är tyvärr slut hos leverantören och kommer inte att erbjudas mer. Om du har tur kanske du kan hitta det på begagnatmarknaden."

Har vi nått så långt att alla i ett IT-projekt har ett användarcentrerat tankesätt kan vi avskaffa dagens användbarhetsarkitekt. I stället kan vi skapa nästa generations användbarhetsarkitekt. En person som inte ägnar sin tid åt att argumentera för sitt eget existensberättigande, inte är ständigt ifrågasatt, inte talar om för utvecklare att det är tillåtet att prata med användarna, inte ägnar dagarna åt att lära ut de mest triviala kunskaperna om gränssnittsdesign och inte benämns som svårsåld kompetens.

I stället ägnar sig denna person åt att tillsammans med kunden och användarna skissa på framtida lösningar. Med sitt kunnande om mänskligt beteende och behov jobbar denne person med strategier för framtida satsningar. 2:a generationens användbarhetsarkitekt är också med i projekten för att koordinera det övergripande användbarhetsarbetet och hålla ett vakande öga på helheten genom hela projektet.

7. REFERENSER

Rhodes, John S (2000). Trouble in Paradise: Problems Facing the Usability Community.

<http://webword.com/moving/death.htm>