



LUND
UNIVERSITY

Underhåll av legacysystem

- verktyg för att hantera användares synpunkter -

Kandidatuppsats, 15 högskolepoäng, SYSK01/INFK03

Framlagd: 16 december 2011

Författare: Tobias Björkgren

Elin Widerberg

Handledare: Hans Lundin

Examinatorer: Agneta Olerup

Claus Persson

Abstrakt

Titel: Underhåll av legacysystem

Författare: Tobias Björkgren
Elin Widerberg

Utgivare: Institutionen för informatik

Handledare: Hans Lundin

Examinatorer: Agneta Olerup
Claus Persson

Publiceringsår: 2011

Uppsattstyp: Kandidatuppsats

Språk: Svenska

Nyckelord: PASiS, legacysystem, gränssnitt, underhåll, användarpåverkan, verktyg

Abstrakt: När ett system utvecklas, utvecklas det i en specifik miljö med specifika krav. Om systemet inte underhålls är systemet konstant medan miljön och kraven ändras. Denna uppsats tar upp legacyproblematiken som uppstår när ett system har åldrats och inte kan möta nya krav. Distribuerad kognition och "Good enough testning" förklarar hur en stor användarbas hanteras och på vilken nivå underhållsarbete läggs med hänsyn till projekttyp och kundbehov. Tietos system används av ett femsiffrigt antal användare med olika bakgrunder. Uppsatsen understryker att en fortsatt utvecklingsprocess inte bör ses som en underhållsprocess då dessa fokuserar på att stödja en existerande funktionalitet. PASiS är ett patientadministrativt system som används i Region Skåne sedan 1970-talet. Det interagerar med bland annat befolkningsregister och laboratoriesystem. Det har utvecklats ett webbaserat gränssnitt som kan användas vid sidan om terminalgränssnittet eftersom båda är baserade på samma bakomliggande system. Slutsatsen är att användare påverkar med hjälp av fokusgrupper vilka filtrerar återkopplingen och sammanfattar det viktiga som då får större genomslagskraft.

Innehåll

1. Inledning.....	4
1.1 Bakgrund	4
1.2 Problemställning.....	5
1.2.1 Forskningsfråga	6
1.3 Syfte	6
1.4 Avgränsningar	6
2. Litteraturoversikt.....	8
2.1 Legacysystem	8
2.1.1 För- och nackdelar.....	8
2.1.2 Undersökning	9
2.2 Feedback.....	9
2.2.1 Participatory design.....	9
2.2.2 Nyckelvärden för användbarhet	10
2.2.3 Positiv effekt	10
2.3 Gränssnittet.....	11
2.3.1 Anpassa för användaren	11
2.3.2 Andra aspekter att ta hänsyn till	12
2.3.3 Utvärdering.....	12
2.4 Good enough testning.....	14
2.4.1 Ramverk	15
2.5 Samband	16
3. Metod	17
3.1 Tillvägagångssätt.....	17
3.2 Intervjuguide	17
3.3 PASiS	18
3.3.1 Tidsaspekten.....	19
3.4 Intervjuer	19
3.4.1 Urval.....	20
3.5 Etik	20
3.6 Validitet och reliabilitet.....	21
3.7 Diskussion	21
3.8 Metoddiskussion.....	21

4. Empiri.....	23
4.1 Intervjusammanfattning	23
4.1.1 Systemet har funnits länge	23
4.1.2 Sammanslagning av flera system till ett.....	23
4.1.3 Vikt vid användares åsikter	23
4.1.4 Verktyg som används	24
4.1.5 Good enough testning.....	24
4.1.6 Omvärldens krav	25
4.1.7 Underhåll	25
4.1.8 Allt måste prioriteras	26
4.1.9 Användaren	26
4.1.10 Utvärdering/testning av systemet	27
5. Diskussion	28
5.1 Användares påverkan	28
5.2 Kvalitet på feedback.....	29
5.3 Jämförelse mellan användbarhetsramverk och Good enough testning	30
5.4 Användarna	31
5.5 Erfarenheter från intervjun	32
6. Slutsats	36
Bilagor.....	38
B.1 Intervjufrågor.....	38
B.2 Intervju.....	39
B.2.1 Intervju med Karin Larsson och Gert Andersson.....	39
B.2.2 Intervju med Lisbeth Hedberg	45
B.2.3 Intervju med Lena Lundgren	47
B.2.4 Intervju med Jan Jörnruud	48
B.3 PASiS.....	50
Källor.....	52

1. Inledning

När ett system utvecklas finns det krav på systemet - dessa krav är systemet tänkt att möta. När omvärlden förändras finns det en möjlighet att listan på krav också förändras. Då utvärderas systemet efter de nya kraven, och resultatet blir då annorlunda. Detta är en sida av de problem som uppkommer när det beslutas att ha kvar ett gammalt system och underhålla det – när ett system funnits kvar länge ökar sannolikheten för förändrade krav. För att motverka problematiken ovan måste också systemet utvecklas - annars ökar klyftan mellan processerna systemet byggdes för att stödja, och dagens processer. En majoritet av utvecklare jobbar med underhåll och om tio år kan så mycket som två tredjedelar jobba med vidareutveckling och underhåll av legacysystem. (Deursen, Klint, Verhoef, 2004)

Det finns verktyg som underlättar vid underhåll av system. Företaget i vår studie har arbetat fram ett eget sätt att arbeta på. När de utvecklade systemet var det inte lika vanligt med kända ramverkslösningar - de utvecklade egna processer istället. På så sätt skiljer de sig från ramverk som till exempel ITIL (Information Technology Infrastructure Library) (Niessink, Vliet, 2002). Detta kommer vi belysa senare i uppsatsen eftersom det är ett välkänt ramverk i dessa sammanhang och existerar i branschen.

Med "legacysystem" menar vi system som varit i drift under en lång tid, ibland över 20 år. De är sammansvetsade med gårdagens arbetsprocesser, vilket leder till svårigheter i dagens arbete. Dock används de fortfarande till de processer de ursprungligen var definierade för. Ofta är systemen också baserade på gammal kod. "Arvet" i detta fall blir då ett svärändrat system som inte stödjer dagens processer. I vissa fall placeras ett nytt gränssnitt utanpå systemet, i andra fall sker det nyutvecklingar. (Hainaut, 2009)

1.1 Bakgrund

När ett legacysystem utvecklades så var de ekonomiskt försvarbara och ingen ifrågasatte funktionaliteten. Men tider förändras och många av dessa system blev föråldrade och dyra att underhålla. De tillgodosåg inte längre användarnas behov. Men en del av dessa legacysystem har företag hållit kvar vid och vidareutvecklat eller till och med bara underhållit då det har investerats pengar i dessa en gång i tiden som räknas in i systemets värde. (Hardin, 2009)

Att byta system direkt kan också bli väldigt kostsamt. Det bästa sättet är att syssla med att undvika risker genom att stegvis satsa på nya lösningar som gör att Tieto undviker risker som legacysystemet orört skulle lett till. Legacysystem är en stor del av budgetar i genomsnitt,

mellan 60 och 80 procent av IT-budgetar läggs på legacysystem. Detta visar på ett av problemen med legacysystem, underhållskostnaden. (Gardner, 2009)

Det är billigare och lättare att förändra ett system i början av utvecklingen än efter årtionden av underhåll. Både för att en tidig förändring kan minska underhållsbehovet men också på grund av att systemets ålder står i proportion med ökade svårigheter att underhålla dessa (Satpathy et. al., 2002)

När ett informationssystem utvecklats och blivit levererat har inte arbetet med det tagit slut. Bland aktiviteter som kan bli aktuella i detta stadium ingår rättning av fel i programkod, ändra tillägg och funktionalitet. Det kan även ingå att anpassa systemet till nya krav eller att förbättra prestanda. Ett samlingsnamn för dessa aktiviteter är underhåll. (Canfora & Cimitile, 2000).

Medan legacyproblematiken inte bara innefattar att ett system föråldrats, kan det även bestå av underhåll av aktiviteter som ska anpassa systemet efter nya krav - krav som uppkommit på grund av ett behov som legacysystemet inte tillfredsställer: legacyproblematiken. Behovet av underhåll uppkommer som en följd av legacyproblematiken. Underhållsprocessen behöver inte börja efter att utvecklingen är klar - underhållet kan börja innan, parallellt med utvecklingen. Detta är i motsats till vattenfallsmodellen, som är en sekventiell utvecklingsprocess som härstammar från tillverkningsprocesser där ändringar i efterhand är kostsamma. Enligt vattenfallsmetoden ska varje steg i utvecklingen vara avklarat innan nästa påbörjas (Sommerville, 2007).

Interaktionsdesign har följande betydelse: Produkter ska designas så att de stödjer de sätt folk kommunicerar på och underlättar deras vardag men även deras arbete (Preece et al., 2002).

För att kunna skapa en bra användarupplevelse så krävs det träning. Med användarupplevelse menas "how a product behaves and is used by people in the real world" (Ibid sid 15). Idag är interaktionsdesign en del av utvecklingsprocessen när ett system eller en hemsida skapas (Mathiassen et al., 2001).

1.2 Problemställning

Vi har valt att undersöka PASiS som är ett sjukhusystem som har användare inom Region Skåne. Systemets vanligaste uppgifter är att sköta de dagliga administrativa rutinerna kring vård av patienter. Förutom det ska det även fungera som ett verksamhets- och ledningsinstru-

ment. Eftersom systemet ska vara en effektiv del i vårdprocessen anser vi att en viktig del då är interaktionsdesign. Detta grundar vi i att fördröjningar i användning av systemet kan ge upphov till fördröjningar i andra områden, där det finns beroenden av data från systemet. Detta kan äventyra verksamheten. Området uppsatsen behandlar är intressant för att interaktion är en viktig del av systemtänk. Vår empiriska och teoretiska undersökning blir mer intressant med en inriktning på legacyproblematik då mycket av den tid som spenderas på ett system är just underhåll. Vår syn på varför vi vill ha med dimensionen med legacyproblematik är för att bättre kunna undersöka hur väl systemet är anpassat för dagens användare, eftersom systemet har funnits i över 30 år. Under systemets livslängd har användare vant sig vid systemet. Detta ställer högre krav och det finns en stor del förutfattade meningar om hur systemet ska fungera och på vilket sätt. Målet med interaktionsdesign är att underlätta för användaren. Eftersom många system som används idag är av en äldre modell där interfacet mot användaren också gjordes när forskningen på interaktion inte kommit så långt som den gjort idag.

1.2.1 Forskningsfråga

Ett av problemen vid vidareutveckling av system är hur krav på funktionalitet ska tillgodoses. Ett annat är att systemet ska tillgodose användares krav och fungera på ett effektivt sätt. Vår övergripande forskningsfråga syftar på att förstå på vilket sätt de beslutande intressenterna avseende PASiS tar intryck av slutanvändarnas feedback och idéer samt hur det sedan inkluderas i underhållsarbetet av ett legacysystem.

Hur förbättras och underhålls ett befintligt gränssnitt i ett gammalt datasystem?

- Vilka verktyg används?
- Hur stor vikt läggs vid användarnas åsikter?

1.3 Syfte

Syftet med denna uppsats är att komma fram till hur ett gammalt datasystem förbättras och underhålls. Även hur användares åsikter kommer in i bilden och hur "Good enough testning" påverkar utvecklingsprocessen. Detta är intressant för att antal legacysystem ständigt ökar eftersom åldrandet aldrig avstannar. Vi inkluderar användares åsikter och dess påverkan för att se vilken positiv eller negativ effekt detta har på vidareutvecklingen av legacysystem.

1.4 Avgränsningar

Alla system som detta företag jobbar med ingår inte i denna uppsats utan det är PASiS som denna uppsats riktar in sig på. Vi beskriver inte hur uppgifter utförs i PASiS eller hur det är

tänkt att framtida versioner av systemet kommer att se ut. Vi avgränsar oss till att fokusera på vidareutveckling och användarpåverkan i den empiriska undersökningen. Det är viktigt att ta med dessa områden eftersom de påverkar varandra. Ett systems förändring påverkar användarupplevelsen på något sätt och återkoppling från användare påverkar hur systemet Vi avgränsar oss till att kolla på verktyg som har med utvärdering att göra. Vi avgränsar oss med att inom området interaktionsdesign så är det gränssnitt vi behandlar.

Vi intervjuar ej användare eftersom denna uppsats behandlar användarpåverkan ur ett utvecklarperspektiv.

2. Litteraturoversikt

Vi börjar med att behandla vad legacysystem är och innehar för egenskaper och kännetecken. Sedan kommer uppsatsen in på feedback och vilken effektiv sådan kan ges beroende på hur mycket feedbacken påverkar. Även gränssnittsteori behandlas och utvärderas. Till sist presenteras "Good enough testning" och några tillhörande effekter. Kapitlet avslutas med en jämförande tabell som visar olika områden.

2.1 Legacysystem

Ett legacysystem kan vara kritiskt för verksamheten. Med detta menas att för att verksamheten ska fungera så måste informationssystemet fungera. Andra kännetecken på ett legacysystem är att det innehåller stora mängder kod, vilket gör det dyrt att ändra om och modernisera, även att informationssystemet har en föråldrad databasmotor, eller ingen alls. Det saknar även flexibilitet om behoven för systemet skulle ändras och även är väldigt sårbart för förändringar i programkoden. (Brodie, 1992)

2.1.1 För- och nackdelar

Det finns både för- och nackdelar med legacysystem. Ofta när dessa system byggdes så var kraven annorlunda från idag och många av applikationerna hade bara för avsikt att köras internt, de skulle inte köras mot externa system. Ett annat problem är att applikationerna inte är flexibla i förhållande till den generella utvecklingen. Det kan bli kritiskt när dessa applikationer körs mot en ny databas. Det saknas även mycket dokumentation över hur dessa applikationer är uppbyggda och hur de hänger samman. I många fall finns inte det företag som från början utvecklade systemet kvar vilket gör att det kan vara svårt att få tag i dokumentation. Om det görs ändringar eller om funktioner skrivs om kan det leda till att delar av systemet kraschar. Det kan även vara så illa att det programmeringsspråk som användes vid utvecklingen inte längre används eller stöds av dagens teknik. Om en ny person ska börja med att underhålla ett legacysystem så kommer det ofta kräva lång inläringstid. (Hardin, 2009)

Men sedan finns det även fördelar med legacyapplikationer vilket delvis förklarar varför organisationer väljer att köra dessa applikationer. De som använder systemet är väl bekanta med dess funktioner och utseende. De vet hur systemet fungerar och hur de ska interagera med det för att nå bästa resultat. Den hård- och mjukvara som används är väl testad och folk litar mycket på denna utrustning. Genom att modernisera systemet till viss del istället för att byta ut det kan nytta dras av den kunskap som utvecklarna hade när systemet utvecklades men även av användarnas synvinkel till en minimal kostnad. (Hardin, 2009)

2.1.2 Undersökning

I början av utvecklingsprocesser är det svårt att veta var undersökningen av verksamheten ska börja någonstans. Det är även svårt att veta vilket tillvägagångssätt som bör användas. Det finns vissa frågor som bör besvaras när ett gränssnitt designas. Först är det viktigt att definiera vilka användarna är och hur de kommer att använda produkten. Det är även viktigt att ta reda på vilken motivation användaren har. Varför används systemet och till vad? Vad är det som gör att användaren inte går en annan väg och använder sig av ett annat angreppssätt? (Sears, A., et. al., 2008).

2.2 Feedback

I detta kapitel kommer vi ta upp att det är viktigt att ta tillvara på feedback. Målet för de flesta system är att stödja sina användare. Om det vid underhållet av ett system ges fokus till frågor rörande användares upplevelser ökar genomströmningen och brister som först uppkommer vid användning blir belysta och underhållstid kan planeras bättre - ur ett användarperspektiv. Om något saknas i ett system kan användare komma med önskemål men dessa kan även komma från inom den egna organisationen som tillverkat systemet.

2.2.1 Participatory design

Det är viktigt att leda feedback till rätt mottagare. Området är "Participatory Design", även kallat användarbaserad design. Evia (2004) genomförde en studie av kommunikationsprocesser och den behandlar kommunikationsprocesser och feedback rörande webbsidor med ett stort antal användare. Dock läggs tyngden inte på antalet användare som helhet utan att det stora antalet ofta leder till mångfald i användarskaran. När ett program släpps kan det vara riktat till en viss målgrupp och släpps då för de användare programmet är designat för. När det gäller webbsidor fungerar det inte på samma sätt. Så fort en webbsida är online kan den nås från alla datorer som har en webbläsare installerad. På så sätt blir det svårare att kontrollera vem som får tillgång och ge en målgruppsanpassad användarupplevelse. Med ett stort antal användare stiger också den mängd feedback som kan fås och i och med att kvantiteten ökar sjunker kvaliteten eftersom det blir svårare att genomsöka den ökande mängden. Det blir ohållbart om gränssnittsutvecklare själva ska sitta och gå igenom berget av feedback och därför får man ta reda på olika sätt att effektivisera hanterandet av feedback. Det föreslås att feedback behandlas i olika lager. Mellan utvecklare och användare finns det experter på användarbaserad design som filtrerar användares åsikter, både gällande kvalitet och till vilken del av systemet det hör och då till rätt person genom analys av ansvarsområde. (Evia, 2004)

2.2.2 Nyckelvärden för användbarhet

Med intåget av webb 2.0 följde det en mängd möjligheter som inte fanns innan. Fokus ligger inte bara på var den tekniska horisonten befinner sig utan hur den nya tekniken kan underlätta för människa-dator-interaktion. Vägen till användarens mål vid en webbinteraktion underlättas genom att kontrollera nyckelvärden för bland annat navigation, funktionalitet och tydlighet. Nielsens tumregler för användarvänlighet och denna slutsats har gemensamma områden som bör belysas. Wu et. al. (2009) för fram “Architectural and Visual Clarity” och “Consistency” medan Nielsen (2005) belyser “Visibility of system status” och “Consistency and standards”. När Nielsen behandlar “Help and documentation” menar Wu et. al. “Online help and user guides”. Varje punkt i den ena samlingen riktlinjer har sin motsvarighet i den andra samlingen. Något som Wu et. al. lyfter fram som skiljer sig är könets betydelse. De menar att vid användartester bör hänsyn tas till kön eftersom olika könstillhörigheter kan uppleva interaktionen olika. (Wu, et al, 2009)

Användbarhet i ett system finns där användaren lätt kan ångra sina val och nå sina mål på ett okomplicerat sätt. Dock finns det mer nyckeltal än hur bra gränssnittet är och hur effektivt det är att navigera vad gäller hur använt ett system blir. En studie som författarna baserar sina resultat på visar att nivån på ett gränssnitts utveckling inte nödvändigtvis speglar användandet av själva gränssnittet. I deras studie var det vanan att jobba på ett visst sätt som trumfade det nyproducerade gränssnittet. Det var såklart inte själva åldern på gränssnittet som belyses här utan fokus ligger mer på att gränssnittet är förbättrat med tanke på användarvänlighet. I deras studie ingick ett gammalt textbaserat legacysystem och ett nyare grafiskt system med en lättöverskådlig navigation. Författarna utförde användartester och kom fram till att det sistnämnda fick högst betyg. Dock visade en uppföljande studie att de flesta fortfarande använde sig av det textbaserade systemet. Något som också spelade in var hur kompetenta och vana testpersonerna var vid de olika systemen - de som kan det gamla legacysystemet bättre känner sig mer säkra med att använda det och ger därmed högre poäng i tester på grund av gammal vana. (Hauck & Weisband, 2002)

2.2.3 Positiv effekt

Det finns tecken på att en positiv effekt kommer att använda sig av förslag från användare. Effekten av detta kallas för “integration av extern feedback” (Franke, Keinz, Schreier, 2008) och syftar på att den feedback användare ger bara tas emot, och påverkar. Även att starta utvecklingen med utgångspunkt från användares idéer har visat sig vara en bra metod. Båda dessa tillvägagångssätt leder till en större chans att den färdiga produkten accepteras av en

större del av användarbasen. De egenskaper som mättes för att komma fram till denna slutsats var hur produkten procentuellt stämde med användarens inre bild av den, varför den är ett attraktivt köp och hur mycket kunden var villig att betala. Efter att de räknat ut sambanden genomfördes det även empiriska studier för att ta reda på om teorin stämde - och fann de teoretiska sambanden ovan. (Franke, Keinz & Schreier, 2008)

2.3 Gränssnittet

Feedback behandlar ofta gränssnittet. Gränssnittet är en brygga som hjälper användaren och systemet att kommunicera med varandra. Ett fokus på uppgifter som ska utföras i systemet bygger på att målen med interaktionen analyserats i förhand. Men ett systems funktionalitet står inte och faller på hur gränssnittet upplevs. Enligt Preece (2002) består processen bakom interaktionsdesign av fyra grundläggande aktiviteter. Dessa aktiviteter är att först ska det identifieras vad som behövs för att skapa en bra användarerfarenhet. Sedan ska det utvecklas olika lösningar som fyller upp dessa krav som framkommit i steg ett. Vidare så ska det byggas modeller av dessa lösningar så de kan testas och till sist utvärderas i hur väl de tillgodosåg användarna och deras upplevelse. När det sista steget är slut så är det meningen att samma process ska börja om från början en gång till; då med de ändringar och åsikter som framkommit första gången. Det händer ofta att utvecklarna måste iterera denna process flera gånger innan produkten är helt klar och samma gäller dess utseende. (Preece, 2002)

2.3.1 Anpassa för användaren

En vidareutveckling av gränssnittet påverkar inte alla delar av systemet. Vid underhåll av legacysystem kategoriseras användningsområdena och deras krav. Funktioner delas upp för att ta reda på vilka som påverkar gränssnittet och vilka som påverkar systemet även om detta sker i kulisserna. Ett webbaserat system där serveråtkomst sker genom webbläsaren gör att webbläsaren då blir klienten. I efterhand sker en jämförelse mellan det nya webbgränssnittet och legacysystemets gränssnitt för att säkerställa fortsatt kvalitet. Med kvalitet menas bland annat förmåga att möta kraven från kravspecifikationen och användaren. (de Camargo & Penteado, 2003)

Den funktionella designen sätter fokus på att göra det enkelt och behagligt att utföra de uppgifter som leder användaren närmare sitt mål. Detaljerad design är det som beskriver alla regler för komponenternas beteende så som stil i ledtexter, typsnitt osv. I detaljdesignen anpassas också form av feedback till det mänskliga systemet och användningssituationen exempelvis genom att gruppera och färgsätta innehåll i fönstret/ på sidan. (Ottersten & Berndtsson., 2002)

2.3.2 Andra aspekter att ta hänsyn till

Användbarhet är viktigt och dess nivå är gränssnittet en stor påverkan på. Det finns flera punkter som förtjänar minst lika stor uppmärksamhet vid utveckling och underhåll, några av dessa är prestanda, hur pålitligt systemet är, hur pass skalbart det är för att möta nya krav och hur säkert det är. De anser därför att det är viktigt i olika processer rörande användbarhet att det ständigt tas hänsyn till resten av tidigare nämnda punkter - att det hålls ett helhetsperspektiv under utvecklingen. Genom att stärka bland annat hur skalbart, säkert och pålitligt systemet är så nås indirekt en högre nivå av användbarhet eftersom antal felkällor minskar. (Bass & John, 2001)

2.3.3 Utvärdering

Nielsens Heuristics är 10 punkter som kan användas som ett verktyg när ett system expertutvärderas för att kontrollera att de bibehåller en viss nivå av kvalitet. Skillnaden mellan systemet och omvärlden belyses genom den checklista Jakob Nielsen tagit fram. Genom att utvärdera hur väl interaktion med tänkta användare stämmer överens med punkterna i listan ska skillnaden i interaktion mellan systemet och framtida användare minska. Det som dessa punkter behandlar är bara riktlinjer som behandlar olika saker att tänka när ett gränssnitt utvärderas. Dessa handlar om hur det ska göras så enkelt och smidigt för användaren att navigera i systemet. (Nielsen, 2005)

1. Den dialog som sker ska vara enkel och bara innehålla sådan information som är relevant och den ska komma i en naturlig ordning för användaren.
2. Använd ett språk som användaren är bekant med.
3. Användaren ska inte behöva komma ihåg olika typer av information utan sådant ska följa med genom processen.
4. Liknande knappar ska ha samma funktioner. Även liknande ord bör betyda samma sak.
5. Användaren ska med jämna mellanrum få information om vad som sker.
6. Det är viktigt med en väl markerad och placerad funktion för att avbryta. Det ska även vara möjligt för användaren att ångra sina val och göra om.
7. Att använda sig av genvägar för den vana användaren är ett sätt att effektivisera deras arbete.

8. Informera användaren när fel uppstår genom meddelanden om vad som är fel och förslag på hur detta kan lösas.

9. Försöka se till så att fel inte uppstår.

10. Ha bra dokumentation där användaren kan gå in och läsa om hur denna kan lösa sina problem som uppstår med systemet och även hur det fungerar. (Ottersten & Berndtsson, 2002)

Användartester är ett annat sätt att utvärdera ett system. Dessa tester kan utföras med jämna mellanrum medans en produkt utvecklas I början kan dessa tester göras med prototyper på papper. Senare kan man göra dessa tester på en dator när de är körbara. Om tester görs i tidigt skede och det saknas funktioner så ska det finnas en handledare vars uppgift är att täcka upp där sådan information saknas så att användaren kan slutföra testet. Det finns olika typer av användartest. (Ottersten & Berndtsson, 2002)

1. Strukturerat användartest - Här får användaren ett antal olika uppgifter som denna ska lösa. Under tiden användaren löser dessa så ska han tänka högt. Det ska ske minst fem tillfällen med samma uppgifter och en person varje gång. Det finns olika roller som ska uppfyllas i ett sådant test. Det är dels den som testar, själva användaren. Det ska även finnas en handledare som användaren kan ställa frågor till. Det ska finnas en observatör också som ska skriva vad som händer. Till sist ska det finnas ett vittne och detta är en person som vill veta resultatet av testet, det kan vara exempel beställaren. Ibland finns det inget vittne utan då kan det finnas två observatörer istället.

2. Partest - Här sitter användarna två och två tillsammans och löser olika problem. De diskuterar med varandra om hur uppgiften kan lösas på bästa sätt. Även här finns en handledare närvarande och dennes jobb är att följa diskussionen och om det behövs, leda den rätt om användarna flyter långt iväg i sin diskussion. Denna typ av test är bra för det ger lite mer bakgrundsinformation om det område som produkten ska stödja och hur användarna ser på detta.

3. Walk-up-and-use - Detta är en typ av test som görs på publika produkter där användarna har allt från lite till mycket kunskap och ett sådant system ska kunna användas av alla på ett bra sätt. Här är det användarna själv som ska hitta på olika uppgifter att lösa och samtidigt prata högt om hur de tänkte. Här har handledaren en annorlunda roll för dennes uppgift är att ställa frågor till användaren. I och med att det är användaren själv som kommer på olika upp-

gifter så blir det svårt att dra generella slutsatser från denna typ av test.

(Ottersten & Berndtsson, 2002)

Vid utvärdering av ett legacysystem ligger fokus på vilka områden som är bristfälliga (enligt normen satt vid utvärderingen) - dessa är de som kan förbättras och fördelarna med att underhålla och utveckla systemet. Fokus måste samtidigt ligga på hur dessa förbättringar förhåller sig till helhetsbilden legacysystemet för nuvarande stödjer. Några förbättringar kanske ses som rena vinster av vissa fast det kan hända att de hamnar utanför ramarna. (Brooke & Ramage, 2001)

Det finns vissa frågor som bör besvaras när ett gränssnitt designas. Först är det viktigt att definiera vilka användarna kommer vara och hur de kommer att använda produkten. Det är även viktigt att ta reda på vilken motivation användaren har. Varför just det systemet används systemet och till vad och varför användaren inte går en annan väg och väljer ett annat angreppssätt. (Sears, et. al., 2008).

2.4 Good enough testning

När ett legacysystem utvärderas letas det efter brister som behöver åtgärdas - men att ett system har brister är inte nödvändigtvis en brist. Detta avsnitt behandlar detta synsättet.

Good enough testning är ett kluvet begrepp. Det kan betyda och tolkas som allt från en metod till att förkorta utvecklingstiden till att syfta på utvecklare som vill komma undan med så lite som möjligt (Bach, 1997). Komma undan i detta fall syftar på att sänka kvaliteten på själva programkoden. Men allt är inte negativt. Något som blivit till av forskning i ämnet är några riktlinjer för "stressade systemutvecklare" där huvudmålet är att genom rätt prioriteringar kunna uppnå den tidsvinst good enough testning strävar efter, och samtidigt lägga fokus på det viktiga (vilket varierar från projekt till projekt). Metoden ger lika bra resultat vid analys av vilka delar som ska föras fram som vilka delar vars betydelse bör minska. (Bach, 1997)

Det har forskats fram olika metoder där good enough testning används. Någon som direkt påverkar kvaliteten genom att rikta sig till testare är den Bach (1998) behandlar. Den siktar in sig på frågan "Hur vet jag om jag har använt rätt testmetod tillräckligt mycket?" (Bach, J., 1998). Bach hävdar att testprocessen har som mål att ta fram tillräckligt välgrundade beslut på ett ekonomiskt sätt. En annan noterbar poäng han kommer med är att tidpunkten är viktig att ta i anspråk. Värderingarna om vad som är viktigt att lägga tid på (och därmed ge ökad ekonomisk resurs) måste stå i proportion till i hur situationen utvecklas. (Bach, 1998)

Något som underlättar användandet av good enough testning i utvecklingsprocessen är något så simpelt som en uppskattningsmetod. Programmeraren uppskattar konsekvenserna av vissa metoder. Nödvändigheten i vad de kommer leda till och exempelvis hur många arbetstimmar som krävs. På så sätt minimeras risken att göra mer än vad som krävs. (Howden, 1998)

2.4.1 Ramverk

ITIL (IT Infrastructure Library) är en samling av "best practice" för detta ändamål och det nämns även att dessa "best practice" är vad kunderna ska eftersträva och vad utvecklarna ska sträva efter att nå. Olika områden som behandlas i ITIL-samlingen av ramverk är olika servicebestämmelser och allmänt om infrastruktur, miljöperspektivet är även förekommande och de sista handlar om infrastruktur angående informationsteknologin, såsom byggnaden och kablage. Fokus ligger på den första kategorin ("The Service Support set") som passar vår avgränsning bäst och där ingår sex stycken områden. Dessa områden handlar om allt ifrån konfiguration, support, förändring och kontroll av systemet till planering av oförutsedda händelser. Något som även behandlas är kundsamarbete, testning och att få systemet accepterat av användare. En brist med ITIL är att processsamlingen säger att mycket ska implementeras och användas, men aldrig hur det ska gå till eller vad som ska ingå. ITIL har även svårt att skilja på produktionstjänster och supporttjänster. Det finns även "IT Service Capability Maturity Model" som är en modell som försöker fokusera processer som producerar kvalitet hos IT-tjänster. Som hörs på namnet är det en modell som beskriver mognad och den består av fem steg med gradvis ökande mognad. (Niessink et al., 2000)

Det finns olika ramverk, mönsterlösningar, som används vid utveckling av interaktionsmöjligheter för system och användare – även kallat interaktionsdesign. Även om titeln kan få läsaren att tro att det handlar om mönsterlösningar för interfaces för att slå samman legacysystem kan den faktiskt tolkas som mönsterlösningar för gränssnitt. Några punkter de använder för att motivera användningen av mönsterlösningar för design; att man inte ska dokumentera alla nymodigheter och använda sig av dessa, utan mer att hålla sig till de som fungerar bäst. En annan är att använda det för att förbättra kommunikation mellan olika utvecklare. Det finns väldigt mycket implicit kunskap i systemutvecklingsprocesser och dessa metoder hjälper till att fånga upp den. (Millard, 2006)

2.5 Samband

Denna tabell tog vi fram för att på ett enkelt och lättöverskådligt sätt belysa viktiga termer som har en stark sammankoppling. Varje rad tar upp ett problem, nyckeltal som påverkar problemet och sammankoppling till olika teorier.

Tabell 2.1. Tabell som visar sammankopplingen mellan olika områden som denna uppsats tar upp.

Problem	Nyckeltal	Wu & Nielsen	Good enough testning	ITIL	Feedback
Finns ingen logisk menystruktur	Navigation	“Architectural and Visual Clarity”			X
Tillgodoser inte användarnas behov	Funktionalitet	Knappar ska spegla funktionen, “Architectural and Visual Clarity”		X	X
Otydlighet	Tydlighet	“Visibility of system status”	X		X
Hur ta tillvara feedback?	Stödja feedback (från nedan)				X
Hur ska feedback påverka?	Använda feedback (från nedan)				X
Svårigheter i att underhålla	Underhåll		X	X	
Legacy-systemet ska vara effektivt	Effektivitet				X
Ej flexibelt system	Flexibilitet, möta krav				X
Vad ska undersökas och stödjas?	Användartest, definiera användarna		X		X
Planera underhåll	Användning, feedback		X	X	X
Leda feedback till rätt mottagare	Participatory Design				X
Underlätta utförandet av uppgifter	Funktionell design		X		X
Hur skräddarsy användarupplevelse?	Detaljerad design				X
Se förbi interfacet	Prestanda, pålitlighet.			X	

3. Metod

3.1 Tillvägagångssätt

Med hjälp av vårt teoretiska ramverk skrev vi frågor som hjälpte oss besvara vår forskningsfråga. Vi intervjuade utvalda personer på ett företag och vidare analyserade den information som framkom. Vi använde oss av en gruppintervju men även enskilda intervjuer.

3.2 Intervjuguide

Vi genomförde intervjuer för att få en djupare bild av hur feedback från användare påverkar utvecklarens dagliga arbete. Först ville vi att de intervjuade skulle ha erfarenhet av utveckling och inte bara varit verksamma inom organisationen och frågade därför hur länge de jobbat på Tieto och hur länge de jobbat som utvecklare (se bilaga B.1 för en helhetsbild av frågorna). Sedan använde vi frågan "Vem bestämmer ytterst hur gränssnittet ska se ut?" för att börja undersöka vilka krafter som påverkar; om det är någon högt uppsatt i organisationen eller om det sker påverkan utifrån. Vi ville ta reda på vad det fanns för riktlinjer för gränssnittet eftersom vi under teoristudien fann olika riktlinjer för gränssnittsutveckling, allt från flera liknande Nielsens heuristics till en samling viktiga nyckeltal där det som stack ut var att vana styr (Hauck & Weisband, 2002), förutom de allmänt förekommande: navigation, funktionalitet och tydlighet (Wu et. al., 2009).

Något som påverkar hur mycket användarpåverkan det finns - inte bara att det finns - är hur ofta kontakt med användare sker, Det är även intressant att undersöka om kommunikationen sker direkt mellan utvecklare och användare eller om det finns mellansteg. (Franke, Keinz, Schreier, 2008)

Då denna uppsats handlar om underhåll och verktyg får vi en empirisk syn på användandet av verktyg vid sidan av den teoretiska, därför finns en fråga om detta med. Vi ville fördjupa studien och tog med nackdelar och fördelar.

Ett av målen med intervjun är att ta reda på hur användare kan påverka och ett steg mot detta mål är att undersöka hur processen ser ut för den roll de intervjuade har och att undersöka vilka processer som bör prioriteras. Vi frågade även om viktiga processer vid utveckling av PASiS, för att kunna jämföra med feedbackprocesser. Det är även intressant att ta in vilka verktyg som stödjer detta och vi fortsätter med djupet och tar in frågor rörande för- och nackdelar även här.

Angående jämförelsen mellan feedback- och utvecklingsprocesser tar vi reda på de intervjuades åsikter om huruvida de flyter samman. Att ta reda på om de intervjuade ser "en poäng med att inhämta användarnas åsikter" motverkar det faktum att det är lätt att förbise utvecklarens åsikter och tro att feedback bara är av godo

Behovet av en interaktionsdesigner är inte en självklarhet och därför undersöker vi om de intervjuade känt ett behov av en interaktionsdesigner och varför; fördelar och nackdelar. Även om deras syn på interaktionsorienterad utveckling/underhåll.

Till sist ställer vi en sammanfattande fråga som tar reda på om våra intervjupersoner anser att slutprodukten skulle förändras med tanke på det som sagts i intervjun.

3.3 PASiS

Det system vi har undersökt är ett sjukhusystem som används av sjukhus i Skåne, med ett stort antal användare. Själva syftet med systemet är att underlätta vårdprocessen, med allt det innebär, till exempel att hantera patienters journaler, att skicka ut kallelser eller att hålla reda på personal. Systemets syfte bygger på att dess processer väl stödjer och efterliknar ovanstående. För att senare analysera användningen av systemet vänder vi oss till området interaktionsdesign eftersom det till stor del fokuserar på användbarhet. Gränssnittet har en stor påverkan på användandet av systemet. Detta är ännu en anledning till att ta hjälp av detta forskningsområde. Då är det viktigt att undersöka vilka krav dagens användande ställer på systemet.

Systemet våra intervjupersoner jobbar med är PASiS; vilket står för PatientAdministrativt Stöd i Skåne. PASiS är ett gemensamt system för hela regionen med fokus på att stödja sjukhusens patientadministration. Det är de "dagliga administrativa rutinerna" som huvudfokus ligger på. Ett exempel på vad som behandlas i systemet är tillgänglighet/väntetid avseende patienter och läkare. Detta görs tillgängligt genom att det görs registreringar och avläsningar i en Flödesmodell. Hela vårdprocessen finns med; från det att sjukvården uppmärksammar patientens problem till dess att de sista aktiviteterna för att åtgärda detta har avbrutits. Detta kan sedan analyseras ner till individnivå. PASiS har en stor variation på användarkrets.

PASiS är skräddarsytt för Region Skåne och är inte avsett att säljas till andra landsting. Tio olika sjukhus använder systemet men även primärvården. PASiS kan köras både via ett webbgränssnitt och som terminalsystem. I dessa fall är det i stort sett samma servrar som an-

vänds. PASiS integrerar med flera olika system, bland annat befolkningsregister och laboratoriesystem. (se bilaga B3)

Enligt vår empiriska studie (Se bilaga B2.1) har en webbdesigner som uppgift att ta fram ett webbgränssnitt till PASiS, och det projektet är i skrivande stund förbi utvecklingsfasen och i användning. Webbgränssnittet (som kallas webbPASiS) har fått stort genomslag men trots detta har programversionen av PASiS en trogen användarskara.

3.3.1 Tidsaspekten

Som nämdes i gruppintervjun har systemet funnits i över 30 år, sedan sent 1970-tal. Från början var PASiS enbart ett terminalsystem och utvecklades av IT-avdelningen på MAS. Systemet spred sig till resten av Skåne när de 3 sjukhusområdena slogs ihop. Det har funnits sedan sent 1970-tal. Från början var det uppbyggt av vanliga filer med nyckelsystem. Det skulle dröja till det blev SQL-baserat.

3.4 Intervjuer

Sättet vi samlade in vår empiri var med hjälp av en gruppintervju och sen tre stycken individuella intervjuer. Något som talar för användandet av gruppintervjuer är när erfarenheter hos individer behandlas. Ibland kanske de inte kan sätta ord på något förhållande. Om någon annan i gruppen berättar något som påminner om hur de känner det kan de genast hålla med. Det är även ett bra sätt att upptäcka om det finns skillnader mellan rollerna som var med i intervjun. (Jacobsen, 2002)

Gruppintervjuer är lämpliga när målet är att få mer kunskap om ett fenomen, men även när ett specifik/avgränsat område ska diskuteras. Det finns en risk att intervjun spårar ur om området är brett. Gruppintervjuer fungerar bäst när målet är att få fram individers synpunkter och åsikter om ett speciellt förhållande, till exempel en speciell händelse. (Jacobsen, 2002). Eftersom de vi valde att intervjua tillsammans har jobbat tillsammans så länge och båda jobbat med PASiS sen det nästan var nytt så fick vi en intervju där vi fick svar på de frågor vi ställde och de kompletterade varandra på ett bra sätt. Detta passade bra för de roller, en som hade mycket kontakt med kunden och skötte det administrativa, en som var den mer ansvariga och sen de som jobbade med PASiS varje dag, vi skulle undersöka, och en positiv effekt var som ovan att skillnader och likheter blir mycket tydligare i en gruppintervju. Vi intervjuade den ansvariga för systemet, och den ansvariga för gränssnittet Vi valde att intervjua två stycken personer

som jobbar på Tieto (som är leverantör av PASiS). Dessa har jobbat på företaget sedan 1980-talet så vi ansåg att de var väl insatta i hur arbetet fungerade.

Vi använde oss av en bandspelare på gruppintervjun. Enligt Jacobsen (2002) är det bra att använda sig av en bandspelare för då underlättas ögonkontakt med den som intervjuas. En nackdel med att spela in kan vara att en del reagerar negativt på att spelas in. En annan nackdel kan vara att den som intervjuar låter bli att anteckna stödpunkter vilket kan göra det svårt att hitta specifik information senare. En tredje nackdel är att det kan lätt bli ett avbrott i intervjun när bandet tar slut och måste bytas. Vid riktig otur så strejkar bandspelaren och hela intervjun går förlorad. Men att spela in intervjun bör man ändå göras så fort tillfälle ges.

Vi planerade att hålla oss till relevanta frågor men med några inledande frågor bara för att komma igång. Detta då en intervju inte bör vara längre än två timmar då det finns risk för att båda parter blir trötta. Korta intervjuer (mindre än 30 min) är för korta då det är svårt att komma in på djupet av ämnet. Korta intervjuer kan användas om det är mer konkreta frågor som inte kräver så mycket diskussion. (Jacobsen, D.I, 2002)

3.4.1 Urval

Vi valde att intervjua folk på ett företag som heter Tieto och det är detta företaget som utvecklar och underhåller PASiS. De vi valde ut till vår gruppintervju är de två som jobbat längst med PASiS och som fortfarande arbetar med underhåll. En annan anledning till att vi också valde dessa två är att den ena är ansvarig för PASiS och har mycket kontakt med både kunden men även en del kontakt med användarna och den andra jobbar mycket med gränssnittet men har även kontakt med både kunden och användaren. Vi valde även ut de rollerna eftersom de kan ha olika syn på frågorna, och olika synsätt på systemet.

Vi valde även ut 4 andra personer som jobbar med PASiS och är väl insatta i det dagliga underhållet för att få ett annat perspektiv på arbetet. Det kan ju variera lite för vilket befattning de har inom företaget hur de dagliga rutinerna ser ut och hur synen på arbetet är.

3.5 Etik

De vi intervjuade i gruppintervjun fick veta vad den information som de gav oss kom att användas till och i vilket syfte. Vi kommer att spela in hela intervjun men bara efter att vi frågat de deltagande om detta var något som de kände sig bekväma med. När det kommer till de övriga som deltog i undersökningen så kommer även dem att vara informerade om vad syftet är med undersökningen och vad materialet ska användas till. (Jacobsen, 2002).

3.6 Validitet och reliabilitet

Vi valde att intervjua 6 olika personer, dock var det bara vid fem av dessa vi fick fram ett användbart resultat. För att få bra underlag och kunna styrka vad som sades på intervjuerna så behövs det minst fyra olika personer att intervjua (Martin, 2005). Vi hade helst velat intervjua fler personer men det var svårt att få tag i personer som kunde tänkas ställa upp på detta och som hade tillräckligt med kunskap om lite olika områden som kunde passa in. Vissa personer var så specialiserade på just sitt område så det hade varit svårt att få fram en helhetsbild från dessa. Vi valde även personer som jobbat på Tieto en längre tid som varit med under själva utvecklingen när PASiS var nytt.

3.7 Diskussion

När det kommer till diskussionen väger vi teori från litteraturstudien mot den empiri som vi får fram. Detta gör vi genom att hitta likheter och skillnader och sedan diskutera runt detta. Bara för att empiri och litteratur säger olika saker så betyder inte det att någon har fel. Dock så säger flera källor samma sak så verkar det mer trovärdigt. Vi redogör för de resultat vi kommer fram till i förhållande till vår forskningsfråga och diskutera kring dessa genom att väga de mot vårt teoretiska ramverk. Vi hoppas på att på så sätt komma fram till en slutsats som baseras på både tidigare forskning från vår teoridel och från den empiri som kommer fram under intervjuerna. Tillslut så görs en sammanfattning av uppsatsen och därefter kommer en slutsats som baseras på vad som kommer fram under de olika kapitlen. (Jacobsen, 2002).

3.8 Metoddiskussion

Vi har använt oss av både teoretiska och empiriska angreppssätt för att inhämta information. Vad gäller intervjumetodik använde vi oss av både enskilda och gruppintervjuer. Nackdelen med att inrikta sig på en teoristudie är att de fakta som är mest förekommande är om generella slutsatser om hur processer används. Nackdelen med att basera sin forskning på en intervju är ett ensidigt resultat (Jacobsen, 2002). Saker ses bara från en synvinkel. Om dessa metoder kombineras så kanske nackdelarna inte upplevs som lika stora. Området som undersöks ses så ur flera synvinklar generella slutsatser undviks genom att få specifik information ut ur intervjun.

Vi har även använt oss av källor från olika årtal, för att minska påverkan från rådande trender. Vissa år är några teorier extra populära och om vi bara hade undersökt ett sådant år hade vårt resultat blivit väldigt färgat av detta. Vi har även både källor i tryckt form och i elektronisk

form. Den trycka formen anses vara mer tillförlitlig medan den elektroniska ofta är mer uppdaterat.

Eftersom vi hade en gruppintervju på grund av att de två olika rollerna kunde komma fram till olika saker fick vi en mer nyanserad bild. Dock var de ofta överens. Men det behöver inte vara negativt och är inte heller negativt. Det visar bara på att det fanns ett samspel mellan rollerna i de olika frågorna: att de hade förståelse för varandra. De är bra insatta med vad den andra jobbar med och att de får samma information från beställaren och användarna då de inte alltid träffade samma personer vid mötena om vidareutvecklingen

Något annat som var slående var hur väl vi hittade källor som stämde överens som underlag till intervjun. Det var inte bara att vi satt och letade efter bekräftelse på att den teori vi plockat ut stämmer, och att sedan kunna göra en enkel jämförelse. Vi fick även höra saker vi inte läst någonstans, vilket såklart var det mest värdefulla. Detta eftersom det fördjupade vår bild av situationen. Vi valde att intervjua båda rollerna samtidigt eftersom att då få en mer nyanserad bild, inte bara en roll separat för sig. Vi skulle ju göra på ett liknande sätt i teorienomgången, att jämföra källor och bland ihop, så lika bra att använda samma metod på intervjun.

4. Empiri

4.1 Intervjusammanfattning

I detta kapitel kommer vi gå igenom områden av vikt för vår diskussion och sortera empiriska fynd under respektive kategori.

4.1.1 Systemet har funnits länge

I början av en intervju berättas det att PASiS har funnits i ungefär 30 år, vilket gör det ännu intressantare, eftersom det då blir ett sant legacysystem. Det har funnits med länge och det har tillkommit stora mängder kunskap om systemutveckling sedan systemet utformades. Dock har de ändrat programvara för databasen sedan det började. De som var med från början var från "den gamla skolan" - vilken vi kan räkna in de vi intervjuade till (se bilaga B.2.4). Att systemet byggde på en databas från början är ett tecken på att det fanns någon teori över att det behövdes struktur redan då. (Se bilaga B.2.1)

4.1.2 Sammanslagning av flera system till ett

Innan PASiS var det ingen brist på sjukhussystem, det fanns 3 olika från början i Skåne, ett för varje landsting. Sedan togs två bort och kvar fanns Malmö Läns landstings system, som då började gälla för hela Skåne, och fick namnet PASiS. De vi intervjuade började komma in i bilden som utvecklare på IT-avdelningen för sjukhussystemet i Malmöhus Län. När Region Skåne och Tieto började samarbete var Region Skåne den formella av parterna.(Se bilaga B.2.1)

4.1.3 Vikt vid användares åsikter

Tieto använder sig av en så kallad fokusgrupp - en grupp slutanvändare som är representativa för målgrupperna – de anser att detta är mer effektivt än att alla slutanvändare ska ge input, och det är mindre risk för att någon användargrupp ska bli överrepresentativ. Ett mindre effektivt alternativ hade varit att alla input skulle inhämtas från varje slutanvändare och detta skulle innebära en risk för att någon användargrupp ska bli överrepresentativ (de som tar sig tid att svara). Det är viktigt med feedback eftersom det ger en positiv påverkan på utvecklingsprocesser (se bilaga B.2.2). Utvecklarna använder sig inte av att kommunicera med slutanvändarna direkt, dock kommunicerar de med fokusgrupperna ibland. Att låta kommunikationen ske via en kontaktperson är inte bara mer praktiskt utan hindrar också att olika användare pratar med olika utvecklare och kommer fram till olika saker. Tieto har implementerat ett "Kundstöd" som tar emot användares kommentarer eller frågor, de är även ISO9000-

certifierade. Utvecklare på Tieto anser att testningen borde vara mer ingående, på det sättet att involvera fler personer som inte är insatta, för att lättare kunna se brister. (Se bilaga B.2.1)

4.1.4 Verktyg som används

Deras syn på ITIL var inte så positiv. När utvecklingen med PASiS började fanns det inga färdiga ramverk för processer att använda sig av så de fick skraddarsy egna processer, som de använder än idag. Det etablerade formaliserade ramverket som finns idag stödjer också önskemål och felanmälningar förutom feedback. Det finns fler anledningar till att respondenterna gärna håller kvar vid sina processer, än använder ITIL som ett verktyg för att hitta nya. En anledning de nämner är att det ofta blir konflikter mellan Tieto och beställaren, i de fall beställaren bestämmer något som inte går att implementera eller deras bild av slutmålet avviker från varandras. Enligt de processer de använder sig av idag kan utvecklarna ta kontakt med Region Skånes kontaktperson direkt, vilket sparar tid. En större formell process här hade mest gjort kommunikationen långsammare. ITIL är inte något som är populärt vid utvecklingen av PASiS. Ingen av de vi pratade med har gått en utbildning i processverktyget. Trots detta hålls en mer formell miljö på Tieto nu än innan PASiS kom in i bilden, vid utvecklingsprocesser och dylikt. (Se bilaga B.2.1)

En viktig process som nämns är att Tieto tillsammans med Region Skåne kommer gemensamt fram till riktlinjer för kommande års förändringar. Här kan det finnas påverkan av kommande politiska beslut inom Regionen som påverkar systemet och uppföljningssystem. (Se bilaga B.2.1)

PASiS är ett interaktivt system som användaren inte behöver uppdatera. Detta genom att de tar fram "Service Packs" minst 6 gånger om året, som installeras på en central stordator som ger effekt på systemet direkt. Dock sker det oftare allteftersom fel upptäcks. Testning sker alltid innan nya ändringar tas i bruk. Användarna har mycket lättare för att kommentera gränssnittet än bakomliggande funktioner. Det leder till att Tieto måste iterera fler tester för att testa funktionalitet. (Se bilaga B.2.1)

4.1.5 Good enough testning

Det är Region Skåne som är beställaren för PASiS och det är de som bestämmer och tar beslut om hur utvecklingen ska gå till. Dock är de mottagliga för förslag, och deras bestämmande går mest ut på att tycka till om prototyper de får. Tieto har insett att det krävs en speciell kompetens för att hantera dessa förslag. Därför beslutade de att anställa en interaktionsdesig-

ner som är helt inriktad på interaktionsdesign. Att personen saknar programmeringskunskap leder till att vissa förslag blir svåra att implementera rent praktiskt. Denna roll är tänkt att fylla ut de svagheter som teknikerna lider av när de ska hantera kundens förslag. Tieto måste följa beställaren. Men det kan bli kompromisser om de vill ha något som inte fungerar. Då frågar utvecklarna kontaktpersonen direkt (de tjänar tid på detta eftersom de kringgår formella processer som ofta slöar ner kommunikationen, enligt dem själva). Samarbetet mellan Region Skåne och Tieto har en speciell uppbyggnad. De är båda styrda av en budget, och strävar efter att nå en ”good enough”-nivå men dock styrs prioriteringarna av kunden (Region Skåne) i slutändan, det är alltid kunden som bestämmer. Det finns en instans inom Region Skåne som gör de officiella prioriteringarna, ”Programkontoret”. De vi intervjuade angav att de har kontakt med beställaren oftare än vad kontraktet säger. Detta är ett sätt att både ge kunden ett större inflytande, och att se till att utvecklingen inte avviker nämnvärt från kundens bild av den färdiga produkten nämnvärt, i alla fall att sträva efter det. Riktlinjerna för utveckling av gränssnittet sker i samarbete mellan Tieto och Region Skåne - de har bildat en gemensam förvaltningsgrupp med ansvariga för de olika patientkontoren i Skåne. (Se bilaga B.2.1)

4.1.6 Omvärldens krav

De vi intervjuade angav att det viktigaste är att följa Patientdatalagen, och även att hålla koll på politiska beslut och anpassa utvecklingen efter det. Även politiska beslut som påverkar vården, eftersom vården är PASiS så kallade ”problemområde”. PASiS är ett system som ska användas av sjukhus, därför måste själva systemet hålla samma höga sekretess. I systemets gränssnitt finns det inga tydliga tecken på denna sekretess. Det användaren märker efter ett tag är att den måste motivera om den vill få mer information om något än vad dennes konto har tillgång till. Detta leder i sin tur till mer knapptryckningar vilket aldrig är populärt hos PASiS målgrupp. Helst vill de bara trycka på en knapp per åtgärd de vill genomföra, enligt de vi intervjuade. (Se bilaga B.2.1)

4.1.7 Underhåll

Gränssnittet - närmare bestämt PASiS - har utvecklats en hel del på de senaste tio åren. Från början var systemet ett terminalsystem, det vill säga det enda sättet att kommunicera med det var genom textbaserade kommandon. Det övergick sedan till att vara ett webbgränssnitt där dialogen med användaren kan ta nyare former, och bli mer dialogmässig istället för att bestå av envägskommandon. Vissa vi intervjuade nämnde att de har en metod angående införandet av webbgränssnitt. ofta är det så att de utformar ett webbgränssnitt till en existerande applikation, och använder sig av existerande datakällor och bakomliggande arkitektur. Det finns rikt-

linjer inom utvecklingsgruppen på utseendet av knappar och färger eftersom det är viktigt med genomgående tema (se bilaga B.2.3). Dagens användare är inte vana vid terminalsystem eftersom de inte är lika vanliga idag, de är desto mer vana vid webbgränssnitt. En sak i intervjun vi reagerade på var att varken Tieto eller Region Skåne lagt någon vikt vid att bevaka trender inom interaktionsdesign. PASiS har även ett webbgränssnitt som sköts av en webbdesigner hos Tieto, dock har Region Skåne en stor ergonomisk kunskap som utnyttjas vid designen. Något annat som nämns är att det är nödvändigt att någon på kundens sida har en diskussion om viktiga detaljer såsom defaultvärden i olika fält, det kan inte utvecklarna på Tieto veta. Förslag på förändringar i gränssnittet presenteras för förvaltningsgruppen ovan; det är de som beslutar om de ska genomföras. (Se bilaga B.2.1)

4.1.8 Allt måste prioriteras

De viktigaste processerna kan inte sägas vara antingen gränssnitt eller funktionerna. De viktigaste processerna innehåller båda delar. Något som även är viktigt är att hitta de nya behoven hos systemet och användarna (se bilaga B.2.2). Kan även vara viktigt hur deras arbetssituation förändras. Det är mycket lättare att testa gränssnitt än funktioner. För funktioner är det en lång kedja med tester, för gränssnitt är det lätt för användare att direkt säga om de tycker att de ser bra ut eller inte. Något som är den viktigaste anledningen varför testerna utförs, enligt utvecklarna, är att hitta nya behov hos användarna. Deras arbetssituation förändras även, med tanke på nya krav på funktionalitet. (Se bilaga B.2.1)

4.1.9 Användaren

När flera olika system ingår i en användares arbete ska byten mellan dessa ske smärtfritt och utan att användaren ska märka det, enligt utvecklarna vi intervjuade. Det blir mer logiskt med en inloggning för hela systemet, istället för att komma ihåg olika lösenord. Det blir även lättare för användaren. En annan vinst med detta är att målet är att det inte ska bli något dubbelarbete mellan systemen, eftersom det övervakas vilka system som gör vad. (Se bilaga B.2.1)

PASiS idag speglar uppgifterna som fanns på sjukhus förr. Att boka in patienter och hantera detta administrativt till exempel. Förr användes papperslistor men idag är det digitalt på skärmen. Ett exempel på hur olika uppgifter PASiS kan få, och hur mycket problemområdet innehåller är att under hösten – 09 drabbades Sverige av svininfluensan och vaccinationer och kallelser till dessa skulle distribueras, detta var systemet tvunget att hantera. Systemet stödjer elektroniska journaler, men användare/sjukhuspersonal föredrar det fysiska alternativet. (Se bilaga B.2.1)

Om Tieto jämför PASiS med när det var nytt och hur situationen är nu kan slutsatsen dras att det finns mycket formella processer nu, bland annat i utvecklingen. Om alla följer processerna finns det inga tydliga brister, enligt de vi intervjuade: ”De som svarar i telefon (kundtjänst/reception) kanske ibland inte kan svara, men då skickar de frågan vidare, så ingen egentlig brist är det inte...”. (Se bilaga B.2.1)

4.1.10 Utvärdering/testning av systemet

Tester borde genomförts lite hårdare jämfört med hur det har varit fram tills nu. Inte hårt som i att ge kritik om användarna misslyckas under testningen, hårt som i att testa i många fler dimensioner (se bilaga B.2.3). Detta är viktigt eftersom testerna måste ge resultat om hur olika användargrupper påverkas – eftersom PASiS har en målgrupp med stor mångfald. Detta är en stor utmaning, det är svårt att alltid testa för alla målgrupper, ofta blir det en kompromiss. De fortsätter att berätta att PASiS kan ses som en stor kompromiss (mellan användargrupper). En annan lösning kunde ha varit att göra en variant av PASiS för varje användartyp. Vissa vill ha ett mer avancerat interface, vissa vill bara ha ett simpelt för den del de har ansvar för och så vidare. Men det hade lett till omöjlighet att testa. Det hade blivit för omfattande att testa alla varianter mot hela målgruppen. Detta har också att göra med det vi nämnde ovan, att ofta är för få testare involverade, mest på grund av ett pressat tidsschema. (Se bilaga B.2.1)

När vi bad de vi intervjuade att göra en jämförelse mellan enkelhet i användningen av gränssnittet och funktionaliteten kom de fram till samma sak, från olika vinklar. Slutsatsen var att det funktionella måste finnas. En effekt av det är att om något förslag måste skäras ner är det nästan alltid ”trevliga saker som får stryka på foten”. Den vi intervjuade som har en utvecklarroll förkortade debatten om Enkelt kontra Funktionellt till att handla om ett behov av färre knapptryckningar och bra svarstider. Om systemet är enkelt interaktionsmässigt behövs det färre knapptryckningar och om det är bra funktionellt får det även bra svarstider. Användarna ogillar att använda musen för mycket. De tycker inte om att byta fram och tillbaka mellan tangentbord och mus. Alla val ska bestå av knapptryck eller något snabbval enligt feedback. (Se bilaga B.2.1)

5. Diskussion

Det finns mycket för- och nackdelar med att bygga vidare och underhålla ett legacysystem. Allt från krånglig kod till att dokumentation saknas, till att koden till och med inte stöds längre. Men sedan fanns det ju också fördelar så som att användarna är vana vid ett visst system. Systemet gör fortfarande sitt jobb det vill säga stödjer de funktioner och processer som användaren förväntar sig. PASiS är ett typiskt sådant system. Det har funnits i över 30 år och används fortfarande som ett av de större systemen inom Region Skåne. Detta måste innebära att de som bestämmer vilka system som ska användas fortfarande anser att fördelarna överväger nackdelarna. PASiS har dock skrivits om och fortfarande pågår underhåll och ändringar för att göra det bättre för användarna. Att förbättra något behöver inte innebära att det utvecklats till perfektion. I vissa fall räcker det att upprätthålla en underhållsprocess som ser till att systemet håller en jämn användbarhet och stödjer sin användarskara. (Brodie, 1992)

5.1 Användares påverkan

Som vi tog upp i 4.1.8 och 4.1.9 är det en stor del av allt arbete som drivs av användare. Att hävda detta är giltigt då det utgår ifrån att det är utvecklarna som arbetar i projekt som har som mål att förbättra något i systemet. Vad som ska förbättras sker utifrån användarnas perspektiv med fokus på hur systemet används. Eftersom feedback används för att komma fram till förslag på nya förbättringar påverkar således användarna både indirekt genom hur de använder systemet och direkt genom att komma med feedback. (Evia, 2004)

Användare påverkar genom fokusgrupper. Detta kan få användare att känna att de hamnar på ett längre avstånd ifrån utvecklarna. Faktum är att effekten blir det motsatta. Med en fokusgrupp som kategoriserar och prioriterar återkoppling från användare får varje förslag större uppmärksamhet av utvecklarna eftersom kanaliseringen av feedback ökas. Med hjälp av kategoriseringen och prioriteringen av feedback kan återkopplingen lättare kopplas till nyckeltal som då lättare kan kontrolleras och användarens mål uppfylls då lättare. (Wu, et al, 2009)

Tietos utvecklare lyssnar på användaren och det finns en användargrupp som nya, utvecklade idéer bOLLAS med. Användarskaran har ökat sen införandet av webbgränssnitt - det är ett tecken på att ett sådant gränssnitt är mer uppskattat än sin föregångare, eller i alla fall mer lättillgängligt. En orsak kan vara att färre användare hade tillgång till terminalgränssnittet eftersom det webbaserade kan nås från vilken dator som helst med tillgång. Att användarna fick vara med att påverka ligger bakom det ökade antalet användare. (Franke, Keinz & Schreier, 2008)

5.2 Kvalitet på feedback

Denna uppsats pekar på flera ställen på att det är viktigt att rikta sig till rätt publik. Vägen dit är dock fylld med många vägval med allt från användartester till att utgå från användares synpunkter. Men teorin kom fram till ett djupare synsätt; alla olika sorters vägval måste baseras på en grundlig genomgång av vilka användarna är - det är inte bara att börja samla ihop några användare och fråga vad de tycker. (Evia, 2004)

Enligt god undersökningsmetodik är det viktigt att se teorier från olika synvinklar och här finns det tecken på att det finns en bakväg. Om det genomförs tester där slumpvis utvalda användare ingår ger ett sådant test utslag på vilka grupper av användare som har minst problem vid användning av systemet. Sedan kan det dras slutsatser om det är rätt grupper som har lätt för systemet - det vill säga om svårighetsnivån i användandet ligger på rätt nivå. Efter det kan en grundlig analys av användare (som är föreslagen) ta fart. Det kan även vara så att det är de användare som har mest problem med systemet som är aktiva och ger feedback och blir därför en överrepresenterad grupp - egentligen är det inte så många som har problem som det verkar.

Kvaliteten på feedback kan variera från hur utförlig den är till om den pekar på några konkreta punkter eller en allmän känsla ("systemet är trögt att använda" kontra "varje gång jag öppnar patientfönstret tar det 10 sekunder") (Hauck & Weisband, 2002). Ett annat exempel är "det tar lång tid att logga in" kontra "det är onödigt många steg från det att man väljer användarnamn till det att man är inloggad".

Tieto väljer inte bakvägen som nämndes ovan utan det flöde teorin förespråkar - en fokusgrupp som är representativ för tänkta slutanvändare. Då fås ett djupare synsätt av feedbacken baserat på vilka användarna egentligen är.

Enligt oss går det inte att ge något exakt svar på hur värdefull feedback är. Att det är värdefullt gav både teorin och empirin bevis på. Något som teorin lärde oss fast som inte nämndes i empirin var ett annat sätt att tackla feedbackutmaningen. Feedback, även kallat återkoppling, syftar på åsikter om något som har gjorts. Nyutveckling knyts närmare användares behov när den är baserad på feedback. Tidsperioden med förundersökningar som ligger före utvecklingsfasen kan kortas ner om det som undersöks redan kommer ifrån användarförslag.

5.3 Jämförelse mellan användbarhetsramverk och Good enough testning

Arbetsprocessen för våra intervjupersoner liknar den vi har beskrivit för good enough testning (Bach, 1997). De nämner även begreppet under intervjun, utan att vi nämnt det och processerna liknar Nielsens ramverk utan att de kanske själva tänker på det. Eftersom de ramverken innehåller mycket "sunt förnuft" är det lätt att processer stämmer överens med dessa på grund av att ramverken innehåller många bra nyckeltal. Till exempel håller intervjupersoner på att försöka dra ner antal knapptryckningar (per åtgärd) på grund av feedback och det är en användarvänlig åtgärd. Vid en första anblick kan dessa områden ses som helt olika, fast ju mer jämförelser som görs ju fler tecken ser man på att de faktiskt påverkar varandra. Enligt vår litteraturgenomgång finns tecken på att good enough testning lika väl kan appliceras på testning som på utveckling (Bach, 1998). Detta leder i sin tur till att punkter från Nielsens, och flera andras, ramverk kan samlas ihop och prioriteras enligt den modellen. Det är omöjligt att komma med motargument om en utvecklingschef säger "Gör ett system som matchar omvärlden till 100 %". Om urvalet är baserat på en good enough testning-analys hade ett sådant uttalande kanske fått svaret "Då kommer bara den åtgärden använda hälften av projektets resurser, om vi istället minskar på det kravet tjänar vi dessutom på att systemet blir mer lättöverskådligt och lättanvänt".

Den främsta anledningen till att underhållet av ett system går efter metoden good enough testning, som vi kom fram till, är ekonomisk. När underhållsprocesser blir utdragna är det budgeten som påverkas. I detta fall används metoden inte som ett planeringsverktyg utan som ett sätt att bibehålla en hyfsad kvalitet utan att kostnaderna rinner iväg (Howden, 1998). Good enough testning som ett prioriteringsverktyg hjälper med detta med exempelvis vad som kan ignoreras och bör uppmärksammas; sannolikheten för att det som ges resurser ger utdelning och att förhindra att antalet mindre viktiga punkter tar upp fokus från de stora frågorna. (Bach, 1998)

Med hjälp av good enough testning kan självklara saker ses i ett mer pessimistiskt men ändå realistiskt ljus och detta leder till att bra egenskaper kan ställas mot andra bra egenskaper för att sälla bland feedback och förhindra att konsekvenser av vissa önskemål inte förminskar användares totala nytta av ett system, genom att bibehålla en proportionerlig syn. (Bach, 1998)

Vi har även behandlat ramverket ITIL som kan ses som en motsats till good enough testning. Detta eftersom good enough testning syftar på hur bra en process ska utföras, medan ITIL mer syftar på vad som ska göras, och lämnar svaret på "hur?" till den som implementerar ramverket. Detta faktum gör att en jämförelse mellan dessa kan bli svår och är snarare ett tecken på att ett bättre alternativ är att låta de samexistera.

5.4 Användarna

Att involvera användarna gör oftast att produkten anpassas på ett bra sätt till användarna även om inte en specifik typ av användartest används. Även om man inte använder sig av en specifik typ av användartest så är det bra att ta tillvara på deras åsikter. På Tieto säger de att de använder sig av användarna som testpersoner, men de ser det inte som att de använder användarna som ett slags verktyg.

Något som kom fram i intervjun som vi inte stött på i någon teori är att de inte var medvetna om att interaktionsdesign var en stor fråga förrän nyligen. Det kan vara så att användarnas krav på gränssnittet har förändrats då andra program blivit mer anpassade interaktionsmässigt. För ett tag sedan så låg fokus mest på funktionaliteten och att programmen skulle fungera i sin rätta miljö. Dagens krav ser annorlunda ut och även utvecklingsmiljöerna har utvecklats (de Camargo & Pentead, 2003). Dock är det troligt att i utvecklingsprocessen fanns det aktiviteter som någorlunda tänkte ut "vad behövs visas på detta steg för att det ska gå att använda", men inte längre än så. I den teori vi har kommit över har företag alltid räknat med interaktionsdesign som en faktor, dock inte så ofta varit så bra på det. Därför lät det rätt unikt när varken beställare eller leverantör hade tagit med interaktionsdesign som en faktor under flera decennier. Dock använde de sig av feedback även innan.

I teorin har vi nämnt att något som kan skrämma användare är antalet möjligheter. I intervjun får vi reda på att användarna vill ha enknappsgenvägar till allt och då kan detta antas leda till ett mer avancerat system med väldigt många möjligheter. Det finns en teknik som handlar om att implementera genvägar. Genvägar syns inte för den nya användaren, men avancerade användare kan använda en speciell tangentkombination för att till exempel göra val i systemet mer effektivt. Fördelen med detta sätt att angripa det är att vid en första anblick blir inte användaren rädd på grund av komplexiteten som gränssnittet hade uppvisat annars. Dock finns det en viss tröskel användaren måste passera kunskapsmässigt för att förstå och lära sig alla genvägarna. Något som också bör hanteras är vad som händer om en användare skulle ove-

tande aktivera en genväg av misstag. Ett system med få möjligheter kan också upplevas som händelsefattigt och stelt. Den nya användaren kanske letar efter en speciell funktion och är osäker vilken kategori den faller under. I de fallen är det bättre att presentera många möjligheter på en gång. (Ottersten & Berndtsson., 2002)

Uppsatsens teorikapitel tog upp att användare är rädda för att de inte kan hantera alla möjligheter de står inför. Intervjumaterialet och teorikapitlet tog upp att användare inte gillar för många navigationssteg (knapptryckningar), helst vill de med endast en knapp genomföra önskad åtgärd. Om det behövs många knapptryckningar, så växer antalet möjligheter. Antalet möjligheter växer exponentiellt som en funktion av antal knapptryckningar som behövs. Användarens ogillande av knapptryckningar kan härstamma från deras rädsla för att överrumplas av möjligheter (Hauck & Weisband, 2002). Fast å andra sidan om allt ska kunna utföras med en knapptryckning kommer möjligheterna med tangentgenvägar att växa exponentiellt. (Bass & John, 2001)

Det finns olika verktyg att ta hjälp av när ett gränssnitt utvärderas, t.ex. Nielsens heuristics. På Tieto användes inte uttalat dessa även om de gör vissa av de punkter som Nielsens innehåller. På Tieto anser de själva att de försöker göra det så enkelt för användaren att använda men samtidigt har de vissa standarder som de måste följa. Även om de använde sig av olika verktyg när de utvärderade systemet så betyder inte det systemet hade varit bättre för användaren. Verktygen ska bara ses som en hjälp inte något som måste följas till punkt och pricka. (Brooke & Ramage, 2001)

5.5 Erfarenheter från intervjun

Ett resultat från intervjun är att de anser att det är viktigt med beställarkontakt, och att det använder sig av det ofta, även om det inte är i kontraktet. Detta är ett tecken på att de är väldigt angelägna om att användarna ska uppleva resultatet som så lite avvikande som möjligt från det som kommit fram vid kundsamtalet. Vår teoristudie visar att en större överensstämmelse ovan även leder till en större chans att användarna kommer acceptera systemet – vilket leder till en ökande användarbas. Det i sin tur leder till att beställaren får mer “system för pengarna”. Detta stämmer också överens med vad som återfinns i teorin, om att det är viktigt med genomgående beställarkontakt i alla faser. Annars ökar risken för avvikelser mellan resultat och beställarens inre bild av det. Med “beställarens inre bild” menar vi det beställaren muntligen säger vid kundträffarna vad den vill ha och även hur beställaren föreställer sig sa-

ker som Tietos kundkontakt förmedlar, om någon ny funktionalitet ska användas eller om det blivit problem av någon anledning. (Evia, 2004)

För att kunna skapa ett bra användargränssnitt så är det viktigt att hela tiden föra en bra dialog och involvera beställaren och så stor del av slutanvändarna som möjligt. Genom att involvera dessa kan antalet missförstånd minskas och det kan också innebära effektivare sätt ta fram ett bra gränssnitt. När det gäller PASiS så är både beställaren aktivt med i utvecklingsarbetet och användarna får en chans att testa produkten innan den sätts i bruk. Något vi kan se är att eftersom beställaren och användarna är med under utvecklingsarbetet så blir det inte några stora överraskningar i slutändan av projektet. De vet ungefär redan hur produkten ser ut och fungerar. Detta sparar både tid och pengar och ingen blir besviken. (Franke, Keinz, Schreier, 2008)

Utvecklarna bakom PASiS använde inte ITIL eller något liknande ramverk för att strukturera sina processer, de gick en egen väg. Detta eftersom utvecklarna bakom PASiS inte använt sig av den, men ändå jobbar fram många bra processer som fungerar för deras arbete. Det kan förekomma att ett projekt automatiskt rättar sig efter ITIL eftersom metoden har en hög trovärdighet. ITIL används då utan vidare analys, som (om den används) automatiskt ger trovärdighet till ett projekt. En annan synvinkel på föregående påstående är att ITIL är en bra utgångspunkt, för att det är ett ramverk som bidrar med flera bra saker som vi nämnt innan; exempelvis underlag till att planera för kundsamarbete, testning och allmän acceptans av systemet. Det kan bidra till att så många använder sig av processramverket, och något som talar för att det är befogat. Ramverket känns som något som behövs vid en nystart av en verksamhet. Tieto i vårt case har funnits i decennium och alltid gjort på "sitt sätt". I sådana fall har nog ett nytt ramverk svårt att slå sig in, om det inte finns någon tydlig vinst med att göra så. Utvecklare och ansvariga bakom PASiS nämnde att de involverar användarna på ett bra sätt. Eller rättare sagt de processer som används har med användarna i en nyttig omfattning. Men det finns alltid rum för förbättringar. De hade kunnat göra en personlig intervju med varje anställd, men det hade nog tagit tusentals timmar med tanke på den stora användarskaran. Det svåra är att anpassa ansträngningen de lägger på processer rörande användarupplevelsen. Det finns inget rätt sätt - bara det sättet som leder till minst fel på användarupplevelsen eller det sättet som leder till en lagom mängd feedback eller båda eller en kompromiss. (Niessink et al., 2000)

Ett konkret exempel av vikten av att anpassa involveringen av användare är att ju mer ett system stämmer överens med användares förbättringar ju mer använt blir det. Inte bara att

kundföretaget tvingar anställda att använda systemet, utan också att de anställda känner att de vill använda systemet.

Den teknik som systemet först använde sig av är inte samma teknik som används idag, men de hade samma behov av struktur, och ansåg det vara viktigt. Detta kan bero på att syftet med systemet inte har förändrats. Krav på kapacitet stiger allteftersom systemet får fler användare och måste lagra mer information. Och responstiden ökar då om inte bättre och nyare teknik används. Vi har så mycket bättre teknik i dagens samhälle så att fortsätta köra ett system på samma plattform som för tio år sedan accepteras inte av användarna. Eller rättare sagt - den försämrade kvaliteten i användarupplevelsen som är en effekt av detta uppskattas inte av användarna. (Bass & John, 2001)

Saker utvecklarna vi intervjuade nämnde som gör systemet mer lättanvänt är att de har anställt en interaktionsdesigner och att de under intervjun kommer fram till att de helst hade velat genomföra mer tester. Men något att ha i åtanke är att vi inte lever i en perfekt värld. Allt måste ställas i relation till den budget som beställaren har. Några brister som finns är att de egengjorda processerna är beroende av att alla i kedjan kan sin del av arbetet. Om någon brister vid exempelvis kundkontakt så försvinner information som kanske varit användbar vid lösning av problemet. Dock nämns det under intervjuerna att det finns processer för detta också, att de då går tillbaka och kollar på kundrespons och analyserar den igen.

De mönsterlösningar vi berättade om innan som vi läste i "Design Patterns for Wrapping Similar Legacy Systems with Common Service Interfaces" - som först handlade om bästa sättet att slå samman legacysystem - även kunde appliceras på även interfacearbete (Millard, 2006). En slutsats vi kan dra av det, är att strukturen som själva interfacearbetet då skulle struktureras, kan användas för att hantera feedback. Att till exempel inte spara all feedback och dokumentera den, utan mer strukturera den, och spara det som leder till den största förbättringen. Feedback är ju också någonting som kan användas för att ge en extra dimension i kommunikationen mellan utvecklare.

Det finns redan skrivna processer (ungefär som mönsterlösningar för programmering) som hanterar hur feedback och åsikter ska hanteras vid underhåll. Men på Tieto har det arbetats fram en egen process på hur arbetet sköts. Anledningen är att dessa ramverksprocesser har kommit till på senare år medans utvecklingen av PASiS började långt tidigare. De trotsar trenden med att använda allt som heter som en förkortning genom att ha utarbetat en skräd-

darsydd process som växt fram naturligt i företaget. De anser att de har funnit sitt "best practice" på hur det går till och hur de hanterar underhåll så att även användarnas åsikter tas tillvara. Alla skräddarsydda processer som tillhör PASiS handlar inte till hundra procent om vidareutveckling: det är processerna rörande "Kundstöd", men med användning av användarnas feedback kan många utvecklingsidéer födas. Även om många av Tietos processer angående PASiS handlar om support och underhåll är det processer rörande vidareutveckling vi ansåg vara lite viktigare att ta upp, eftersom de passar bäst in i vår avgränsning.

6. Slutsats

En slutsats vi drog utifrån Tietos situation är att lösningen de valt - att kombinera olika typer av gränssnitt - är bra eftersom det gamla gränssnittet inte går förlorat samtidigt som det kan genomgå en förnyelse genom det nya. Dessutom ökar ju acceptansen av systemet hos användarna när gamla möjligheter kvarstår samtidigt som det finns en väg att gå för de som vill förnya sitt arbetssätt. Systemet förlorar på detta sätt ingen användargrupp

Att följa de befintliga verktygen till punkt och pricka är något som inte alltid är så bra. Att använda sig av olika verktyg och sedan omarbete dessa så de passar just det sätt ett företag jobbar på är effektivt. Tieto utvecklade egna arbetsprocesser och har inte upplevt några problem med det. Det visar att ramverk inte är nyckeln till framgång, bara ett hjälpmedel. Vi drog även slutsatsen att ett verktygs fulla potential för ett projekt fås bara när det analyseras tillsammans med kompletterande verktyg. Utförandet påverkar val av arbetssätt.

Vi har funnit att det är viktigt att lägga tyngd bakom hanteringen av feedback och att bevis för detta återfinns i vår empiriska studie. Vi kan även dra slutsatsen att riktlinjer inte behöver hållas, men trots det är de ofta till bra hjälp. Tieto delar dessa värderingar. Det har inte påverkat kunderna negativt; då de i kontakt med Tieto säger att "det flyter på bra", och enligt deras kunder, som inte påverkas negativt av denna värdering, funkar detta bra. Med hjälp av good enough testning analyseras vilken kvalitetsnivå underhållsprocessen ska hålla. En slutsats vi drar är att eftersom våra intervju svar pekar på är att good enough testning används hos Tieto; utvecklarna och kunden har gemensamt kommit fram till en bra nivå att lägga sig på.

Det vi kommit fram till är att det verktyg som används mest är användartester. Även att våra intervju personer har processer som liknar Nielsens ramverk utan att det har varit ett aktivt val. Eftersom de ramverken innehåller mycket "sunt förnuft" är det lätt att processer stämmer överens med dessa på grund av att ramverken innehåller många bra nyckeltal. Till exempel håller intervju personer på att försöka dra ner antal knapptryckningar (per åtgärd) på grund av feedback och det är en användarvänlig åtgärd. Att använda sig av användartester är ett bra verktyg för då är det lättare att under utvecklingen ta del av åsikter från användarna och på så sätt redan från början ha dessa åsikter i åtanke.

Att förbättra något behöver inte innebära att det utvecklats till perfektion. I vissa fall räcker det att upprätthålla en underhållsprocess som ser till att systemet håller en jämn användbarhet och stödjer sin användarskara. Med hjälp av "good enough testning" analyseras vilken kvali-

tetsnivå underhållsprocessen ska hålla. En slutsats vi från våra intervjuer pekar på är att utvecklarna och kunden gemensamt kommit fram till en bra nivå att lägga sig på. Den processen liknar den vi har beskrivit för good enough testning i teoriavsnittet samt att de nämner begreppet under intervjun, utan att vi nämnt det.

I denna uppsats har vi kommit fram till att användares åsikter inte ignoreras, ofta sker det ett nära samtal mellan användare och kontaktpersonen(-er) hos leverantören. Ibland sker det enligt en definierad process, ibland tar utvecklare och användare direkt kontakt med varandra. Denna metod har sina fördelar: respons fås/ges snabbare och produkten har en större chans att förankra sig hos användaren. Det oanvända upplevs ofta som mer främmande och konstigare än något som användaren följt i utvecklingen och kunnat påverka. Användares åsikter tas till vara på ett produktivt sätt - däribland integration av extern feedback.

Vår slutsats är att det inte finns en bästa lösning; som alltid beror ju lösningens värde på den specifika situationen. I detta fall beror det på mer än så eftersom den smidigaste lösningen för alla parter i vår undersökning är att ge alternativ. Det finns fall där den bästa lösningen inte är varken eller utan både och. Fast vår slutsats är inte att det bästa är att välja en lösning, det tankesättet måste bortses från. Med hjälp av en fokusgrupp kanaliseras användarnas återkoppling på vidareutvecklingar som i sin tur filtreras genom olika metoder och påverkar utvecklare. Det är då valet av lösning inträffar; som inte är ett aktivt val av utvecklare utan egentligen påverkad från början av användare.

Bilagor

B.1 Intervjufrågor

Sitter och utvecklar och underhåller PASiS hos utvecklaren:

- Hur länge har du jobbat här?
- Hur länge har du varit utvecklare?
- Vem bestämmer ytterst hur gränssnittet ska se ut?
- Vad finns det för riktlinjer för utveckling av gränssnittet?
- Vem har kontakt med användare och hur ofta sker det?
- Vilka verktyg använder ni er av för att göra gränssnittet?
- Vad är nackdelen med dessa?
- Vad är fördelen jämfört med andra?
- Hur ser processen ut (för din roll) som används för att ta tillvara feedback?
- Finns det några speciellt viktiga processer?
- Används det instrument för detta?
- Vad har du för åsikter om de instrument som används?
- Finns det några brister enligt dig? (Och vad?)
- Finns det några speciellt viktiga processer vid utveckling av PASiS?
- Vad har du för åsikter om dessa?
- Finns det några brister enligt dig? (Och vad?)
- Upplever du att det vi pratat om, processer för att ta tillvara feedback och utvecklingsprocesser flyter ihop? Varför/Varför inte?
- Ser ni en poäng med att inhämta användarnas åsikter?
- Varför såg ni ett behov av en interaktionsdesigner?
- Har ni funderat på att införa en interaktionsorienterad utveckling?
- Vilka problem ser ni med detta?
- Och skulle slutprodukten bli en annan om detta infördes?

B.2 Intervju

B.2.1 Intervju med Karin Larsson och Gert Andersson

2010-05-19

Vi: "Legacy systems, och med legacy systems menar vi ett system som funnits under lång tid som man underhåller och som fortfarande tjänar sitt syfte, och vad vi förstår så har PASiS funnits sen 80-talet..."

Karin: "Det har funnits sen tidigt 70tal, men det finns inte så mycket kvar av det, det har byggts på och databasen har ändrats, och det har utvecklats under en tid..."

Vi: "ja det är det vi ska kolla på, hur det har utvecklats..."

Karin: "Ja."

Vi: "vi kan väl börja med... Hur länge har ni jobbat med PASiS? Om ni har funnits med sen början eller bara de senaste åren"

Karin: "vi har ju inte varit med sen allra första början, men vi har hållit på med detta länge, vi har hållit på med detta i tjugo år ungefär"

Gert: "Ja man kan ju ta historian också. Man brukar ju säga att PASiS... Att vi har kört med det nonstop, plattformen har funnits sedan 70talet. Men då körde man inte SQL alltså. Då var det Scribe, vanliga filer med nyckelsystem. Och sen har det vidareutvecklats och idag kör vi SQL-databaser, men fortfarande samma plattform, så det har hängt med på den utvecklingen. Jag kommer knappt ihåg när jag började... 87... började jag där. I den gruppen. Sen så har jag jobbat med olika saker men har alltid hållit på med PASiS, någon gång höll jag på med tandvård ett tag, men har ändå jobbat med PASiS en del, men du började något år senare?"

Karin: "88 började jag, det är ju därför jag vet, att du redan hade varit där något år."

Gert: "Så jag har varit där ett tag"

Vi: "Hur ofta har ni kontakt med typ beställaren, Region Skåne...?"

Karin: "Jag har kontakt med beställaren varje vecka. Regelbundna träffar, sådana som är formella, har vi en gång i månaden. Men kontakt med den ansvarige beställaren har jag i princip varje vecka."

Vi: "Får ni någon respons från användarna eller sköts de genom region Skåne?"

Båda: "Det beror på vad det är"

Karin: "Det händer att vi får det från slutanvändaren men det brukar gå via deras kontaktpersoner, som kan vara både slutanvändare själva eller så är de tillsatta som rena systemförvaltare så de jobbar bara med förvaltningen"

Vi: "Sen har vi typ... I maintenancedimensionen finns det olika verktyg/processer såsom ITIL och IT Service Capability Maturity Model... Använder ni den typen av verktyg?"

Båda: "Det vet vi inte..."

Gert: "ITIL beskriver ju processer, det har ju kommit från de senare åren, mer formellt och så. Vi har ett Kundstöd som tar emot först, de löser många fall själva, sen får vi då... vi är en baksupport för

dom, vi får fall de inte kan fixa. De är ISO9000 certifierade, så där är de beskrivit vilken process man jobbar där. De tittar på sådana processer som ITIL osv. Vi har mest ärendeprocesser som kommer in mellan vår support och dom då. Då får vi felanmälan, det finns ett nummer dem kan ringa.”

Karin: "Det finns en formaliserad gång, sen vad processen heter vet jag inte.”

Gert: "Sen vet jag inte om ITIL omfattar utveckling också. För vi har ju vad det är för gång vi ska använda när vi sitter i produktion och så. Det finns också beskrivit. Och för kontakter osv.”

Karin: "Det finns en sådan formaliserad gång för önskemål och felanmälningar. Det kan vara större eller mindre tillägg, hur de kommer in och behandlas och godkänns och hur vi gör. Dom testas hos oss och hos kunden. Nu är detta inte en ”produkt”, men vi kan säga att vi har 6 releaser om året.”

Gert: "Det är bestämt vissa dagar vi ska flytta över grejer till ... det är väl vad kunden bestämt... vilka dagar... Om det är fel och grejer så får vi ju lägga över oftare, såklart. Detta finns också beskrivit mer formellt nu”

Karin: "Man kan se det som att vi lägger ut ett Service Pack”.

Gert: "Det lägger vi på ett ställe, sen slår det igenom direkt. Eller det är inget som går ut egentligen, vi kör ju på en mainframe”.

Vi: "Hur mycket direktiv får ni från beställaren? Hur ni måste utforma systemet, mest gränssnittet...”

Karin: "De är med i processen, så rätt så mycket är det dem som bestämmer det. Dom på Regionen är inte superbra på detta. Vi får förslag och så gör vi prototyper som de får titta på och så ändrar vi i dem. Ja, vi provar om det funkar som dem tycker. Nu ska vi ta in en professionell interaktionsdesigner. Madeleine, som är en riktig specialist på detta.”

Gert: "Hon är expert på gränssnitt, hon vet ingenting om programmering. Hon önskar hejvilt, sen är det någon annan som får göra det. Det är inte alltid det funkar, och det tar tid.”

Karin: "Det är nytt för oss att ta in en konsult utifrån, vare sig för Tieto eller för Region Skåne. Hon kommer ju in som konsult, för att titta på det här med gränssnittet...”

Gert: "Ja hon har ju vettiga åsikter. När vi har gjort de här grejerna har det inte varit någon som tittat på det. Det har varit rent tekniskt. Då har vi inte tänkt på användaren”

Vi: "Du menar som det var innan ...? ”

Gert: "Ja vi är ju tekniker, vi har ingen känsla för hur... En tekniker tänker ’vi lägger ut det där’ och tänker inte på varför, det kanske är bättre med en viss ordning, så att det flyter på bra... Användare har fått tycka till innan så lite feedback har vi fått”

Karin: "O ja, dem kan ju sin verksamhet. I vilken ordning bilder ska komma, hur fält ska ligga, om det ska vara defaultvärden på saker, det är ju dem som kan tala om för oss hur det ska se ut. När det gäller färger och sådant så har vi från början haft en kille som när det gäller webbgränssnitt som kan det där. Vad man ser och vad som är vilsamt för ögat. Inom Regionen är de rätt duktiga på det ergonomiska. Inte för litet eller för blurriga färger. Eftersom det är en så stor arbetsplats så finns det ju en del som är lite handikappade som det också ska fungera för”.

Vi: "Hur påverkar sekretess- och personuppgiftslagen utformandet av gränssnittet? Hur pass mycket

det styr...”

Gert: "Ja... Alltså vi har räknat på det, det kommer påverka. Hur mycket man får se. Man måste ha en relation till patienten för att få se. ”

Karin: "Man ska ha en vårdrelation, men på frågan så hela vårt system är ju genomsyrat av sekretess- och patientdatalag. Men om själva gränssnittet som sådant, där är någon sak vi ska plocka bort, där man ser onödigt mycket, enligt patientdatalagen, men annars är det nog mer sekretesssystemet som styr, mer än gränssnittet”

Gert: "Enligt patientdatalagen så kan de redan strypa behörigheten, de hade kunnat göra det imorgon om de velat. Men jag vet inte om det kommer påverka... I patientdatalagen står det väl inget om vad systemet ska visa eller så.”

Karin: "Jo det handlar mer om åtkomst och loggning”

Vi: "Jo i patientdatalagen stod det mest om åtkomst”

Karin: "Men gränssnittet som sådan ser jag inte blir så påverkad av patientdatalagen. Det är mest att du måste göra aktiva val vad du vill se och får inte se för mycket åt gången. Kunnat motivera varför man vill se mer, då ska det loggas. Ska man ta det fullt ut så kan man se att då får man ett gränssnitt med fler knapptryckningar och det tycker inte användarna alls om”.

Gert: "Det är inte alls populärt, med fler knapptryckningar. Som blir en effekt av valen man måste göra...”

Vi: "Hur mycket måste ni rätta er efter beställaren? Om ni anser att dem är fel ute etc. Om ni tar in en specialist och den anser att det är det bästa och beställaren inte håller med”.

Gert: "Vi lyssnar nog mer på användaren.”

Karin: "Det är beställarens system. Det är Region Skånes system. Så säger dem att de inte vill ha det på ett visst sätt så måste vi följa, även om vi tycker att det är klumpigt eller fult eller inte optimalt”

Vi: "Om det inte går att göra som beställaren vill ha det, gör ni nästan som de vill ha det eller tar ni kontakt med dem och kommer fram till något nytt? Rent allmänt.”

Karin: "Vi har ofta diskussioner under gången och det stöter vi på ofta att de vill saker som vi säger att det inte går att göra eller. Ofta går det att göra men det skulle kosta mer än de är beredda att betala. Och då har vi alltid en diskussion och då får det bli kompromisser”.

Gert: "Det är inte så att vi väntar tills officiella möten om vi har frågor utan vi ställer dem direkt till beställaren eller någon annan, annars går man bara runt och slösar tid.”

Karin: "Eftersom det är ett så stort system har vi en som är officiell beställare, den som har pengarna, men sen finns det alltid en användargrupp, om man utvecklar ett tillägg, som man testat det på. De bollar man sina idéer och problem med.”

Vi: "Hur mycket av gränssnittet har utvecklats de senaste tio åren? Fått fram ny information om hur det ska se ut etc...”

Gert: "Hur länge har vi haft webben... 10 år”

Karin: "Webben har vi haft i 10 år. För 10 år sen var det terminalbaserat, alltså teckensystem. Så under

de senaste 10 åren har de fått ett webbgränssnitt som kunden är mycket mer förtjust i. Vi har fått mycket mer användare sen vi började använda oss av det gränssnittet. Och dem har varit med och påverkat det”.

Gert: "Vi kan väl säga att varken vi eller kunden har varit och bevakat trender inom GUI:n”

Karin: "Det stora steget är ju att vi gått ifrån terminalsystem till webb, sen de senaste trenderna. Krav, ergonomiska krav, om man kollar på om systemet skulle se ut som en iPod, sådant har vi ju inte bevakat. Det är därför vi har tagit in ett riktigt proffs på detta nu som kommer titta på webbgränssnittet, eventuellt ändra lite granna.”

Vi: "Låter ni användarna testa både funktioner och gränssnitt innan ni sätter det i bruk?”

Gert: "Ja”

Vi: "Alltid?”

Gert: "Ja”

Karin: "Enligt de här testrapporterna ska de testa både funktionalitet och gränssnitt”

Gert: "Inget ska gå vidare innan det blivit testat, speciellt de större grejerna”

Vi: "Och processer tar tillvara på detta... åsikter och tankar”

Karin: "Det finns ... När vi kommit så långt att de börjar testa, eftersom vi hela tiden har det fortlöpande så säger de aldrig 'OJ blev det såhär?' utan det är mer kontrollerat. När de väl får testa är de ganska väl medvetna om vad de får. Sen är mycket duktigare på att testa gränssnittet än funktionaliteten. Gränssnittet är ju det man kan se. Vilken ordning knappar ska ligga, vilken färg de ska ha, hur rubrikerna ska se ut. De är de mer glada för att göra än att testa komplicerade funktioner. Där det en lång kedja av tester för att det ska vara fullgott”

Gert: "Det är ju processer, de ska ju godkänna det innan slutresultatet. Planerade ändringar och beställningar de har gjort. ”

Vi: "Vilka anser ni vara de viktigaste processerna vid arbetet med PASiS, alltså ... Vidareutveckla gränssnitt eller funktioner osv.?”

Gert: "Det är väl kombinerat. Det är nog både och. Det beror på vad man kommer fram till och vad den här konsulten säger. Sen är det kanske vissa funktioner man också måste titta på...”

Karin: "Vissa funktioner har kanske blivit förlegade, sen det viktigaste för vår verksamhet är att följa sjukvården som sådan, att IT stödjer de nya behoven som kommer. Att vi kan möta nya behov, och vi har gjort integration med andra system och applikationer. Det är det som är mycket nuförtiden. Det ska vara en inloggning och så kommer man åt allt och så ska man inte märka att man byter system. Labbsystemet eller patientadministration etc.”

Gert: "Det är mycket sådant. Och det viktiga är att stödja verksamheten. Sedan kommer det nationella med vårdkontakter etc. Att om man listar om sig så ska sjukvårdens system uppdateras. Ibland verkar det som att vissa tycker det är tekniskt kul och driver på utvecklingen att allt ska vara sammankopplat. Det tenderar att bara tjäna ett tekniskt syfte. Är kanske en generationsfråga. Sedan kommer det tjänster som BasInformationsTjänst med mera. Det skulle inte kunden märka, skulle inte påverka verksamhet-

en.”

Karin: "Koordination med andra tjänster och se om det finns behov. Om det kommer nya patienttaxor måste vi anpassa systemet till dom.”

Gert: "Det är politiska beslut. Också system där man bokar in automatiskt.”

Karin: "Ja och i höstas skulle alla kallas för vaccination. De skulle registreras i PASiS.”

Gert: "Remissystemet har vi också. Rutiner för att kalla alla.”

Karin: "Det med familjeläkare är också integrerat i PASiS, trots att det är ett annat system.”

Gert: "Vad var det, 1,5 milj. kallelser?”

Vi: "Anser ni att PASiS fortfarande fyller samma syfte nu som när det utvecklades?”

Gert: "Det har det väl, det är ju ungefär samma funktioner, är där någonting man har tagit bort?”

Karin: "Nä. Dem saker som PASiS bestod av från början finns kvar, de centrala funktionerna. Sätt upp patienter på väntelista, boka in dem, producera listor, nu finns de på skärmen. Markera att ed genomfört sina besök, besöken ska generera ekonomi”.

Gert: "Finns väntelistor på slutet vård också.”

Karin: "Så skriver man in vilka som är på väntelista där och som besöker så ska det bli ekonomi av det också. Patienten betalar en del.”

Gert: "Om du ligger och får vård, och är färdig, men måste vara kvar, då tar kommunen hand om det. Samvården. Skåne har handlat upp ett system där som ska införas, men tror inte att det påverkar PASiS. Vi har en del fakturering där. Även om det görs nya system eller så, så överlappar det inte. Det är väl samma, men journalbiten finns fortfarande. ”

Karin: "Journalbiten är på papper fortfarande. De är dock digitaliserade, men finns vissa saker i pappersjournalen fortfarande. Osäkert vad de har det och varför. Finns en modul i PASiS som har hand om det. Ska på ett möte imorgon, och vi bollar fortfarande med ändringar i hur de har pappersjournalerna. ”

Vi: "Undrar om ni (på mötena) har haft diskussionen om att ha kvar pappersjournalerna eller varför de har kvar dem?”

Gert: "Den diskussionen har inte vi. Men de ska köra med elektroniska journaler”

Karin: "De kör med elektroniska överallt, men jag vet inte varför de egentligen har kvar pappersjournaler. Jag vet inte egentligen varför de har pappers, men kanske pga lagar att de måste sparas en viss tid. Men bra fråga!!”

Vi: "Processerna ni använder er av, tycker ni dem är bra? Har ni hittat brister i dem? Även ITIL osv.”

Gert: "Vi kanske använder någon form av ITIL men vi har ju inte använt de begreppen.”

Vi: "Men ni använder andra...”

Gert: "Vi har ju inte gått någon ITIL-utbildning. Just nu har det varit mest med SCRUM, sådant med veckomöten. Vi använder inte ITIL som så, vi använder något som vuxit fram under 20 år”

Vi: "Så ni har byggt upp en egen process...”

Gert: "Ja om hur vi jobbar. Vi träffas varje vecka vi inom arbetsgruppen och går genom vad som har

gjorts vad som ska göras och vem som ska göra vad. Vi får även genom olika problem som uppstår och hur vi löst dessa. Har vi haft möten med kund så går vi genom vad som sagt och bestämts och vilka förändringar som måste ske.”

Karin: "Jo de har vi, och den har ju blivit mer och mer formaliserad. Systemet har vuxit, kunderna... fler kontaktpersoner. Vi hade lösare ärendehantering för 20 år sen”

Gert: "När vi började med PASiS var Skåne indelat i tre län. Karin och jag jobbade på Malmö stads system. Sen slog man ihop de tre systemen så det blev ett. Då hade man tre system och skulle ha ett. Då använde vi malmösystemet. Innan det var vi mer som en IT-avdelning till Malmö sjukhus.”

Karin: "Vi jobbade direkt med dem (Malmö sjukhus) och det handlade bara om det när vi började. Dom som jobbade mot Malmö stads landsting hade säkert kontaktpersoner mot där vi var mer informellt”

Gert: "Vi kan ju säga att när vi kom in på kommunen så blev det en dialog, i samband med det som man gjorde PASiS... Där hade man de där rutinerna X antal dagar per år som man hade möten.”

Karin: "Ja vi hade mer att vi la ut nya versioner när vi var färdiga med dom.”

Gert: "Vi hade nog en demo vi släppte, det var mycket med agilt då, snabbare puckar då. Då var vi mer som en avdelning inom samma organisation. Idag är det mer formellt, och det är mer hur Tieto är. Vi tillhör ett annat bolag, så mer formaliserat.”

Karin: "Vi har ju formella grejer i kontakten med kunden.”

Vi: "Tycker ni att det finns brister i dessa processer ni har nämnt? Som kan förbättras?”

Gert: "Nej det fungerar väl bra om alla följer dem.”

Vi: "Anser ni att ni får ut det bästa av det och tar tillvara och gör det som är bäst för användaren...?”

Gert: "Det kan vara ibland när man får in ärenden, de där nere som svarar i telefon kanske inte kan ställa rätt frågor. Så man får inte fullständiga där”

Karin: "Testningen kunde blivit bättre, även om vi försöker följa... Kunde blivit hårdare och mer omfattande innan man levererar. Eftersom det är en sådan stor organisation. Skulle vi kunnat göra det bättre? Ja det hade vi... Så att det skulle passa olika användargrupper. Men facit är en stor KOMPRI-OMISS. Det ska passa alla, inte skraddarsytt för någon”

Vi: "Det funkar för alla...”

Karin: "Ja t.ex. vid bokningen jobbar man olika om man jobbar inom psykiatrin eller barnavården eller så, så skulle man behöva olika professionalitet, fast sådant har vi fått ge avkall på. Det är inget som passar alla, men hade varit väldigt bra för vissa.”

Gert: "Man kan ju ha massa varianter (på användargränssnittet), men då blir testerna mycket mer omfattande. Då får man multiplicera antalet tester med antalet varianter man har.”

Karin: "När man har ett sådant här system som levt under många år då är det väl lite också att man får hitta en 'good enough testning'-nivå, de har inte en budget att göra allt vad de velat göra med systemet så prioriteringar får styra”

Vi: "Är det ni eller dem som leder prioriteringarna?”

Karin: "Det är ju dem som har pengarna. Vi försöker att uppnå ett visst partnerskap. Rådgöra och, ja vi får lämna tidsuppskattning och så får vi diskutera fram och tillbaka vad som ska prioriteras. Vad som är viktigast? Sköts numera centralt på Region Skåne. Prioriterar över alla ITsystem."

Gert: "Det är kunden ju."

Gert: "Programkontoret som prioriterar över ITsystem, även någon instans över som prioriterar över Programkontoret."

Vi: "Vi kan väl avsluta med att fråga om ni anser att det är viktigare att systemet stödjer användarnas behov funktionellt eller att användaren tycker om att jobba i systemet? Om det är enkelt att använda. Vart ligger prioriteringen? Alltså vad ni tycker."

Gert: "Det måste väl vara..."

Karin: "Det är svårt att säga antingen eller, för funktionaliteten måste finnas där. Sen kanske sådana här som kanske gör det, att lägga pengar på att gränssnitt ska se trevligt ut har fått stryka på foten. Något som vi brottats med att det måste finnas bra svarstider. Långa tider trivs man inte med."

Gert: "Så lite knapptryckningar som möjligt. Inte använda musen. Jag vet inte om det är en generationsfråga. När det ska knappas vill man inte hålla på och skifta för mkt, då tar det mer tid."

Karin: "Allt ska gå och göra via snabbval och knapptryckningar. Och det är ju för att de ska gilla att jobba med det, inget som bidrar till någon ökad funktionalitet. Att det blir behändigare och smidigare"

Gert: "Jag tror ändå, om du jämför Terminal och Webb, Webb är trevligare att jobba med för användare och har besparingspotential. De flesta använder ju sig av browsers idag."

Karin: "Det är mer likt andra system idag, då får vi nya användare som inte använt de tidigare, då känner de igen sig, det blir mer intuitivt. Man slipper kunna alla funktionsknappar som finns i sambands-system."

Gert: "Vissa funktioner kan man köra både i terminal och webb. Vissa funktioner var bara översatta rakt av (från webb), så de kanske man borde lagt ner lite mer tid på. Det är också en prioriteringsfråga, alltid en kund som bestämmer, även om vi kan lägga förslag."

Vi: "Är det något mer du vill säga?"

Vi: "Inte som jag kommer på nu, vi har fått reda på så mycket..."

Vi: "Ja då är vi väl klara nu..."

Karin: "Om det är något mer ni kommer på så får ni gärna maila över."

Gert: "Då är vi klara."

B.2.2 Intervju med Lisbeth Hedberg

Sitter och utvecklar och underhåller PASiS hos utvecklaren:

– **Hur länge har du jobbat här?**

Sedan 1984, alltså 26 år

– **Hur länge har du varit utvecklare?**

Sedan 1977, alltså 33 år

– Vem bestämmer ytterst hur gränssnittet ska se ut?

Vi ger förslag på hur vi tycker det ska se ut. Detta presenteras för kunden (förvaltningsgrupp på Region Skåne) som ger synpunkter.

– Vad finns det för riktlinjer för utveckling av gränssnittet?

Hittills har vi ganska ofta gjort ett nytt webbgränssnitt av gamla applikationer, vilket innebär att vi använder de gamla serverna för kontroller och databashantering. Vi styrs därför till en del av det som redan finns. De riktlinjer som finns har vi bestämt inom gruppen, t ex utseende på knappar, alt-keys som ska vara enhetlig, färger o dyl.

– Vem har kontakt med användare och hur ofta sker det?

Vi har en PASiS-ansvarig (Karin Larsson) som träffar förvaltningsgruppen en gång per månad. Alla i gruppen har daglig kontakt angående olika ärenden t ex frågor, fel osv.

– Vilka verktyg använder ni er av för att göra gränssnittet?

Jag använder Eclipse.

o Vad är nackdelen med dessa?

Det finns ingen syntax-kontroll på Java-script.

o Vad är fördelen jämfört med andra?

Annars tycker jag syntax-kontrollen är bra.

– Hur ser processen ut (för din roll) som används för att ta tillvara feedback?

När vi har utvecklat en beställning flyttar vi över denna till ett DEMO-system där användarna (förvaltningsgruppen) kan testa, utvärdera och ha synpunkter.

– Finns det några speciellt viktiga processer?

Testning, feedback.

– Används det instrument för detta?

Vi gör testprotokoll och testfall som användarna ska godkänna.

– Vad har du för åsikter om de instrument som används?

De fungerar bra. De skulle kanske kompletteras med andra instrument som utformats av användarna själva.

– Finns det några brister enligt dig? (Och vad?)

Användarna testa bara vad vi föreslår. Det är inte alltid vi vet exakt hur funktionerna används i verksamheten.

– Finns det några speciellt viktiga processer vid utveckling av PASiS?

Testning, leveransprocessen.

– Vad har du för åsikter om dessa?

De är mycket viktiga.

– Finns det några brister enligt dig? (Och vad?)

Testning borde i allmänhet göras mer ingående, kanske av flera personer inte bara av den som gjort ändringen. Ofta är det tidsbrist som styr här.

– **Upplever du att det vi pratat om, processer för att ta tillvara feedback och utvecklingsprocesser flyter ihop? Varför/Varför inte?**

Ja, feedback är en del av utvecklingsprocessen.

– **Ser ni en poäng med att inhämta användarnas åsikter?**

Ja, det är mycket viktigt eftersom det är de som ska jobba med det verktyg som vi utvecklar.

– **Varför såg ni ett behov av en interaktionsdesigner?**

Det skulle vara bra att diskutera design av bilder o dyl. med ett proffs eftersom det ibland kan vara svårt att få till det eftersom vi inte har denna utbildning.

Dessutom är det viktigt att systemet ser enhetligt ut.

– **Har ni funderat på att införa en interaktionsorienterad utveckling?**

Vet inte.

– **Vilka problem ser ni med detta?**

Gränssnittet har varit sig likt länge; om man ändrar för mycket är det risk att användarna blir förvirrade.

– **Och skulle slutprodukten bli en annan om detta infördes?**

Vet inte, användarna är nöjda och vana vid det nuvarande gränssnittet.

B.2.3 Intervju med Lena Lundgren

Sitter och utvecklar och underhåller PASiS hos utvecklaren:

– **Hur länge har du jobbat här?**

Sedan 1982

– **Hur länge har du varit utvecklare?**

Sedan 1982

– **Vem bestämmer ytterst hur gränssnittet ska se ut?**

Ibland jag, ibland kunden.

– **Vad finns det för riktlinjer för utveckling av gränssnittet?**

Bakgrund, stylesheet, uppbyggnad av bilden ska vara likartat.

– **Vem har kontakt med användare och hur ofta sker det?**

Jag, så ofta det behövs.

– **Vilka verktyg använder ni er av för att göra gränssnittet?**

Netbeans och notepad.

– **Vad är nackdelen med dessa?**

Netbeans promptar för mycket.

– **Vad är fördelen jämfört med andra?**

Har inte testat andra.

– **Hur ser processen ut (för din roll) som används för att ta tillvara feedback?**

Vet ej.

– **Finns det några speciellt viktiga processer?**

Vet ej.

– **Används det instrument för detta?**

Vet ej.

– **Vad har du för åsikter om de instrument som används?**

•- **Finns det några brister enligt dig? (Och vad?)**

– **Finns det några speciellt viktiga processer vid utveckling av PASiS?**

Testning.

– **Vad har du för åsikter om dessa?**

• **Finns det några brister enligt dig? (Och vad?)**

Ibland är testarna lata och testar bara de lättaste sakerna och då är det svårt för oss att inte styra användaren.

– **Upplever du att det vi pratat om, processer för att ta tillvara feedback och utvecklingsprocesser flyter ihop? Varför/Varför inte?**

Vet ej.

– **Ser ni en poäng med att inhämta användarnas åsikter?**

Ja, för att kunna bli bättre.

– **Varför såg ni ett behov av en interaktionsdesigner?**

För att kunskap saknas för att göra ett så bra jobb angående detta som möjligt

– **Har ni funderat på att införa en interaktionsorienterad utveckling?**

Nej.

– **Vilka problem ser ni med detta?**

Risken finns att det blir stor omställning för både oss och kunden vilket skapar förvirring

– **Och skulle slutprodukten bli en annan om detta infördes? Hoppas inte, då vi har kontakt med kunden och försöker utveckla enligt deras önskemål.**

B.2.4 Intervju med Jan Jörnrud

Sitter och utvecklar och underhåller PASiS hos utvecklaren:

– **Hur länge har du jobbat här?**

Dels 1976-1979, dels 1986-2010, närmare 30 år

– **Hur länge har du varit utvecklare?**

Sedan 1973

– **Vem bestämmer ytterst hur gränssnittet ska se ut?**

Region Skåne, som är användare av systemet.

– **Vad finns det för riktlinjer för utveckling av gränssnittet?**

Görs i samråd mellan Tietos ansvarige och Region Skånes ansvarige, i en gemensam förvaltningsgrupp tillsammans med ansvariga för resp. patientkontor

inom Region Skåne.

– **Vem har kontakt med användare och hur ofta sker det?**

Det sker en löpande kontakt med Tietos personal och personer inom Region Skåne.

– **Vilka verktyg använder ni er av för att göra gränssnittet?**

Använder HP:s verktyg

– **Vad är nackdelen med dessa?**

•- **Vad är fördelen jämfört med andra?**

– **Hur ser processen ut (för din roll) som används för att ta tillvara feedback?**

Vi har ingen uttalad process för detta, kontaktvägarna är korta och snabba.

– **Finns det några speciellt viktiga processer?**

En viktig process är att se vad som kommer att dra upp riktlinjerna för närmast kommande års förändringar.

– **Används det instrument för detta?**

Nej

– **Vad har du för åsikter om de instrument som används?**

• **Finns det några brister enligt dig? (Och vad?)**

– **Finns det några speciellt viktiga processer vid utveckling av PASiS?**

En viktig process för utvecklingen av PASiS är att tillsammans med Region Skåne slå fast riktlinjerna för närmast kommande års förändringar. Kan ha att göra med kommande politiska beslut som kommer att tas inom Region Skåne och som har påverkan på PASiS och uppföljningssystemen från PASiS.

– **Vad har du för åsikter om dessa?**

• **Finns det några brister enligt dig? (Och vad?)**

– **Upplever du att det vi pratat om, processer för att ta tillvara feedback och utvecklingsprocesser flyter ihop? Varför/Varför inte?**

– **Ser ni en poäng med att inhämta användarnas åsikter?**

Hoppas att användarnas åsikter har tagits tillvara innan vi påbörjar vårt egentliga förändringsarbete

– **Varför såg ni ett behov av en interaktionsdesigner?**

Vet inte vad man lägger i ordet ”interaktionsdesign”?

– **Har ni funderat på att införa en interaktionsorienterad utveckling?**

Har inte hört talas om detta! Vad lägger man i uttrycket ”interaktionsorienterad utveckling”? Skall användarna ”sitta i utvecklarnas knä” under denna process?

– **Vilka problem ser ni med detta?**

Inga

– **Och skulle slutprodukten bli en annan om detta infördes?**

Tror jag inte

B.3 PASiS

PASiS – Region Skånes patientadministration

tieto.com

2011-05-17

PASiS – Patient Administrativt System i Skåne

PASiS

PASiS är Region Skånes patientadministrativa system. PASiS är utvecklat av Tieto på uppdrag av Region Skåne (och dess föregångare Malmö kommun, Malmöhusläns landsting och Kristianstads läns landsting). Utveckling och vidareutveckling har pågått under de senaste 30 åren. Idag finns i stort sett ingenting kvar utav det som gjordes före 1990. PASiS är inte en produkt som är avsedd att säljas till andra landsting utan det är en skräddarsydd lösning för Region Skåne.

PASiS används på Skånes 10 sjukhus och inom primärvården (vårdcentraler). Under den senaste månaden (april 2011) har 13600 olika personer använt PASiS. Dessa användare är alla sjukvårdspersonal. Det finns ännu ingen funktion där patienter kan använda PASiS.

Teknisk lösning

PASiS kan både köras som terminalsystem och som webssystem. Dessa två olika gränssnitt använder till stor del samma bakomliggande servers. De programmeringsspråk som använts är Cobol, Screencobol, Java, Javascript och HTML. Databasen är en SQL-databas. Databasen har med åren blivit ganska stor. Den innehåller t ex över 100 miljoner patientbesök.

Funktionalitet

I PASiS finns över 100 olika funktioner (bilder). Exempel på det som kan göras i PASiS är:

- Schemaläggning av vårdgivare och mottagningsrum
- Väntelistor till öppen och sluten vård
- Registrering av remisser
- Bokning av patienter
- Inskrivning/Utskrivning av inneliggande patienter
- Besöksregistrering med kassafunktion
- Registrering av diagnoser och åtgärder vid besök/inneliggande
- Uttag av papperslistor och excelfiler för vidare bearbetning

I PASiS finns ett antal integrationer med andra system. Exempel:

- Befolkningsregister för att alltid har korrekta person-/adressuppgifter
- Laboratoriesystem för att hämta provsvar
- Journalsystem för att slippa dubbelregistrera uppgifter
- Ekonomisystem för debitering
- Överföring av patientuppgifter till medicinsk utrustning som t ex ultraljudapparater
- Datalager som används för uppföljning och årsredovisningar

Källor

Bach J. (1997): Good enough quality: beyond the buzzword, IEEE, *Computer*, Vol, 30 Nr 8, pp 96-98.

Bach J. (1998): A framework for Good enough testing, IEEE, *Computer* Vol 31, No 10, pp 124-126.

Bass L. & John B. E. (2001): Supporting Usability Through Software Architecture, IEEE, *Computer*, Vol 34, No 10, pp 113-115

Berndtsson J. & Ottersten I. (2002): *Användbarhet i praktiken*, Studentlitteratur, Lund.

Brodie M. (1992): *The promise of distributed computing and the challenges of legacy information systems*, Springer, Berlin.

Brooke, C. & Ramage, M. (2001): Organisational scenarios and legacy systems. *International Journal of Information Management*, Vol 21, No 5, October 2001, pp 365-384

de Camargo V. V. & Penteadó R. D. (2003): An alternative object oriented reengineering for thin-client environments, IEEE, *Proceedings of the XXIII International Conference of the Chilean Computer Science Society* pp 137-144.

Deursen v. A., Klint P. & Verhoef C. (2004): Research Issues in the Renovation of Legacy Systems. University of Amsterdam.

Evia C. (2004): Quality Over Quantity: A Two-Step Model for Reinforcing User Feedback in Transnational Web-Based Systems Through Participatory Design, IEEE, *Transactions on professional communication*, Vol 47, No 1, pp 71-74

Franke N., Keinz P. & Schreier M. (2008): Complementing Mass Customization Toolkits with User Communities: How Peer Input Improves Customer Self-Design, Blackwell, *Journal of Product Innovation Management*, Vol 25, No 6, pp 546-559

Gardner, D. (2009) *Staying on legacy systems ends up costing IT more.*

<http://www.zdnet.com/blog/gardner/staying-on-legacy-systems-ends-up-costing-it-more/3231> (besökt 2010-09-06)

Hainaut, J.-L. (2009) Legacy and Future of Data Reverse Engineering, IEEE, *2009 16th Working Conference on Reverse Engineering*, pp 4.

Hardin, J. (2009) Modernizing Legacy Systems, Articlesbase.com

<http://www.articlesbase.com/business-articles/modernizing-legacy-systems-1379231.html>

(besökt 2011-11-06)

Hauck R. V. & Weisband S. (2002): When a better interface and easy navigation aren't enough: examining the information architecture in a law enforcement agency, *Journal of the American Society for Information Science and Technology*, Vol 53, No 10

Howden W.E. (1998): High-Assurance Systems Engineering Symposium, IEEE, *High-Assurance Systems Engineering Symposium, 1998. Proceedings. Third IEEE International*, pp 166-175.

Jacobsen D.I. (2002): *Vad, hur och varför?*. Studentlitteratur, Lund

Johnston R., Brignall S. & Fitzgerald L. (2002): *'Good Enough' Performance Measurement: A Trade-Off between Activity and Action*, Palgrave Macmillan Press, Birmingham.

Martin, R. (2005): Validity vs. Reliability: Implications for Management. *Rotman Magazine*

<http://hbr.org/product/validity-vs-reliability-implications-for-managemen/an/ROT011-HCB-ENG>

(besökt 2011-03-23)

Mathiassen L., et al., (2001) *Objektorienterad analys och design*, Studentlitteratur, Lund.

Mcgladrey.com. (2010) *Ignoring the hidden costs of legacy systems could prove costly for midsized businesses*. <http://mcgladrey.com/Information-Technology/Ignoring-the-hidden-costs-of-legacy-systems-could-prove-costly-for-midsized-businesses> (besökt 2010-09-06)

Millard D., Howard Y., Chennupati S., Davis H., Jam E-R., Gilbert L. & Wills G. (2006) Design Patterns for Wrapping Similar Legacy Systems with Common Service Interfaces, IEEE, *2006 European Conference on Web Services*, pp 191-200

Nielsen J. (2005): Ten usability heuristics. http://www.useit.com/papers/heuristic/heuristic_list.html (besökt 2011-11-06)

Niessink F. & van Vliet H. (2000): *Software maintenance from a service perspective*, Wiley, Amsterdam.

Preece, J. (2002) *Interaction Design: Beyond human-computer interaction*, Wiley, USA.

Satpathy M., Siebel N. T. & Rodriguez D. (2002): Maintenance of object oriented systems through re-engineering: a case study, IEEE, *Software Maintenance, 2002. Proceedings. International Conference on*, pp 540-549

Sears A. & Jacko J.A. (2008): *The human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications*, CRC Press, London.

Sommerville, I. (2007) *Software Engineering* 8.

<http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/ElectronicSupplements/LegacySys.pdf>
(besökt 2011-03-23)

Wu Y., Tao Y. & Yang P. (2009): The Discussion on Influence of Website Usability Towards User Acceptability, IEEE, *2009 International Conference on Management and Service Science*, pp 1-4