

Kravhantering med prototyp

Mona Lif

www.logica.se/userexperience

Kravhantering med prototyp - agenda

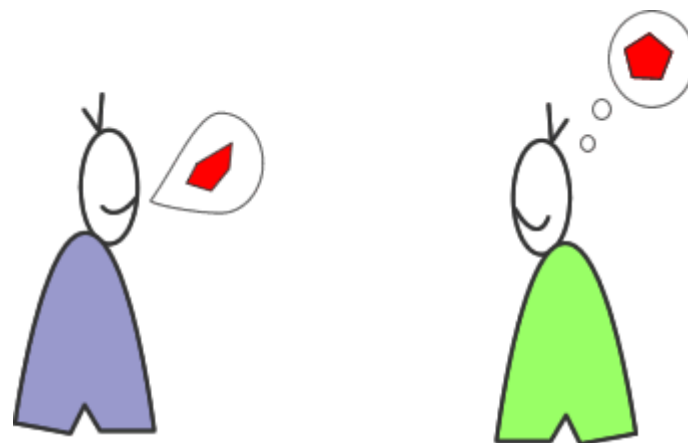
- Fördelar med att använda prototyp
- Hur gör man
- Prototyp vs textuellt beskrivna krav
- Exempel på verktyg för prototyping

Gemensam bild

När man jobbar med utveckling av IT-produkter är det ofta svårt att säkerställa att alla inblandade har samma bild av vad som ska tas fram.

Särskilt innan utvecklingen har på börjats händer det ofta att vi säger och dokumenterar ett krav, som någon annan uppfattar på ett helt annat sätt. Vi har helt enkelt olika inre bilder av vad som ska tas fram.

Om man redan under tidig kravfas arbetar med en prototyp är det mycket större chans att var och ens inre bild av den framtida produkten överensstämmer.



Hitta motstridiga krav

We would like an unspoiled paradise with lots of shops.

Att visualisera kraven i form av en prototyp underlättar även att hitta motstridiga krav.

Text blir det i en prototyp tydligt att allt inte kan placeras på startsidan.



Hitta alla relevanta krav

Krav på vattenkannan:

- Den ska gå att fylla med vatten
- Det ska finnas handtag
- Den ska vara röd som vår profolfärg
- Den ska rymma 1 liter vatten
- Den ska vara tillverkad av ett tåligt material
- Det ska gå att hälla ut vattnet



Det är även viktigt att hitta alla relevanta krav.

Det kan många gånger vara lätt att glömma bort att beskriva krav om man uppfattar som självklara. Men för den som ska bygga applikationen är det inte säkert att det är självklart.

Även sådana hål i kraven kan man hitta om man tar fram en prototyp.

I det här exemplet har jag beskrivit några krav på en vattenkanna. Bilden visar en vattenkanna som uppfyller kraven. Man hade glömt att lägga till det sista kravet "Det ska gå att hälla ut vattnet"

Kunna testa tidigt med användare

En annan mycket bra fördel med att tidigt göra en prototyp är att man kan testa prototypen på användare, och man kan göra det mycket tidigt, till och med innan någon hunnit börja utveckla.

Det är ju med riktiga användare man kan hitta alla oväntade infallsvinklar och behov som man inte alls tänkte på från början.

Man kan även lägga ut en prototyp på t ex en blogg eller liknande för att få kommentarer.



Kommunikation, kommunikation, kommunikation...

Syftet med att ta fram en prototyp handlar om kommunikation.

Inom kravteam

Med styrgruppen

Med projektet

Inom användare

Med kunden

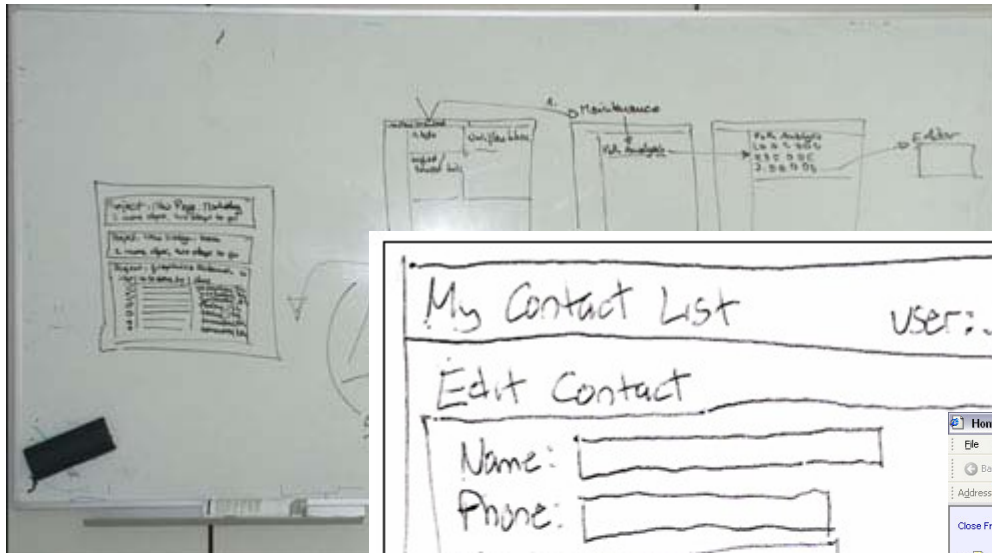
mfl

Så många som möjligt ska ha samma bild av vad som ska byggas

Ju fler som har samma bild, desto mindre risk för obehagliga överraskningar och oriktiga förväntningar.



Hur gör man då?



My Contact List User: John Doe

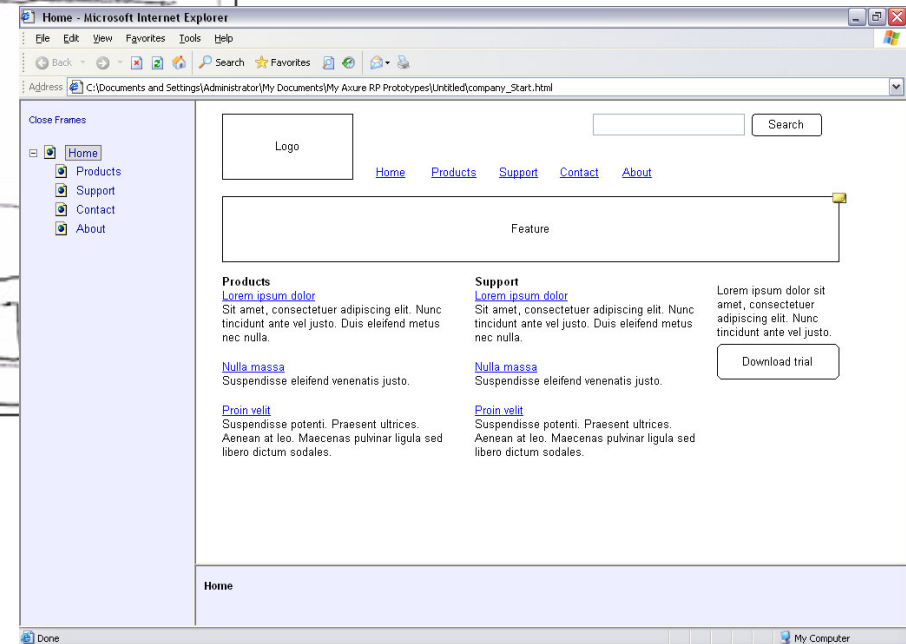
Edit Contact

Name:

Phone:

Email:

Notes:



Hur gör man då?

Man kan säga att man jobbar med prototyp på många olika nivåer. Det viktigaste är att man visualiserar, skissar i någon form.

Hur enkla eller interaktiva prototyperna är beror på när i processen man befinner sig samt vilket behov projektet har.

Krav på dokumentation påverkar även valet av nivå.

Den enklaste formen av skissande är att bara rita på en whiteboard. Nästa steg är att skissa på papper, antingen för hand eller i något enkelt ritprogram.

Om man vill att prototypen ska vara lite mer interaktiv, dvs att man ska kunna klicka sig runt mellan några olika sidor kan man använda något prototypverktyg, t ex Axure.

Om projektet redan valt teknik och man har tillgång till någon som kan tekniken kan man även ta fram en prototyp i rätt teknik, t ex html.

Det viktigaste är att det är lätt att ändra i en prototyp, du ska även kunna hålla flera olika designförslag uppe i olika prototyper, för att en bit in i processen välja spår.

Risker:

- att prototypen blir för "färdig layout" för tidigt
- Man måste ta hänsyn till projektets budget och tekniska förutsättningar, risken finns att man i prototypen skissat på en för komplicerad lösning.

För en enkel applikation kan en prototyp, med kommentarer, ersätta stor del av den textuella kravspecen, men ofta behövs både prototyp och textuell kravspec.

Inled med att definiera vision, mål och de övergripande kraven.

Det viktigaste är att man börjar jobba med prototypen parallellt med att de textuellt beskrivna kraven tas fram. Detta är väldigt viktigt eftersom båda skärningarna prototyp och use case påverkar varandra.

T ex i och med att ett use case visualiseras i en prototyp inser man vilka krav som saknas eller vilka krav som man behöver beskriva tydligare.

Tar man fram use case först, fryser det och sedan realiserar det i en prototyp kommer man stöta på problem när man i prototypen hittar bättre lösningar som inte överensstämmer med det som beskrivits i use case.

Den första prototypen ska tas fram i workshopsform. Det är då viktigt att deltagarna är representativa så att alla krav kommer fram. T ex någon användare, verksamhetsexpert, utvecklare, it-arkitekt, kunden etc. (glöm inte utvecklare!)

Exempel på textuellt beskrivna krav

Söktjänst

1. UC börjar när kunden öppnar startsidan för söktjänsten.
2. Kunden skriver in ett ord i sökfältet.
Systemet kontrollerar det angivna ordet.
3. Systemet presenterar en lista med sökträffar
4. Kunden väljer att titta på en av sökträffarna och UC avslutas.

Alternativflöde

- Avancerad sök...

Frågeställningar som kan komma upp då en prototyp tas fram

- Ska verkligen vanlig sök och avancerad sök vara helt olika ingångar?
 - T ex att söka på bilder, olika språk redan i vanlig sök?
- Ska systemet föreslå sökträffar redan medan kunden skriver?
- Hur ska listan med sökträffar se ut, url, sidans titel, etc?

Detta är ett exempel på hur krav beskrivna i ett use case lämnar en hel del frågor olösta. Det kan många gånger vara dessa frågor som är skillnaden mellan en bra produkt och en dålig.

Tekniker för prototyp

- Papper/whiteboard och penna
- Verktyg
 - Axure
 - Visio
 - Powerpoint
 - Flash
 - Fireworks
 - Illustrator
 - Expression (Blend)
 - Thermo (Adobe)
 - Etc
- Koda direkt

Tack!

mona.lif@logica.com

www.logica.se/userexperience