

Utförare
Andreas Hägglund
Godkännes

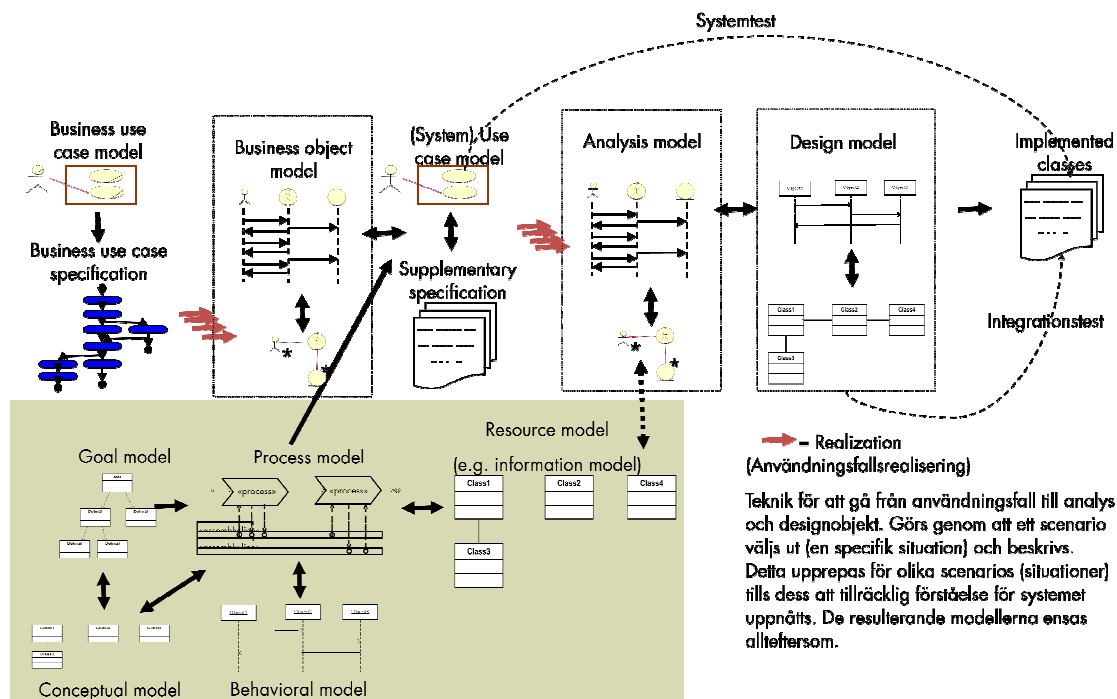
Datum
2009-11-21 17:51:00
Dokumentbeteckning

Utgåva

Från verksamhetsprocess till testat system

1 Inledning

Detta dokument beskriver på två sätt hur man kan gå från verksamhetsmodell till testat system.



Ovanstående bild sammanfattar arbetet från verksamhetsmodellering till systemutveckling. Den visar dessutom två olika sätt att verksamhetsmodellera på, traditionell modellering/-astrakanmodellering (det nedre gråa fältet) och verksamhetsmodellering à la RUP. Från och med kravmodellen skiljer sig inte de två ansatserna åt.

2 Verksamhetsmodellering à l'Astrakan

Vid astrakanmodellering fokuserar man på fyra olika modeller, mål-, begrepp-, process- och informationsmodellen. Dessa kan dock kompletteras med andra modeller, såsom intressentmodeller, vid behov. Ofta inleder man arbetet med målmodellering för att sedan gå vidare till processmodellering (för att beskriva vad man gör för att nå målen). Under

Utförare
Andreas Hägglund
Godkännes

Datum
2009-11-21 17:51:00
Dokumentbeteckning

Utgåva

processmodellering identifierar man sedan ett otal informationsobjekt som man hanterar i processerna och roller som utför arbetet. Dessa strukturerar man sedan vid tillfälle i en informationsmodell. Begreppsmodelleringen gör man vid behov, så fort ett behov utkristalliserar sig.

3 Verksamhetsmodellering à la RUP

Vid verksamhetsmodellering enligt RUP så arbetar man på ett liknande sätt, med liknande begrepp, som vid systemutveckling. Arbetet tar sin utgångspunkt i verksamhetens intressenter (verksamhetsaktörer, eng. business actors) och man frågar sig vad de vill ha ut av verksamheten. Dvs vilka verksamhetsanvändningsfall/processer (eng. business use cases) som finns i verksamheten. Respektive verksamhetsanvändningsfall beskriver man sedan med hjälp av ett aktivitetsdiagram. Under arbetet med aktivitetsdiagrammet identifierar man roller (eng. business workers) och resurser (eng. business objects) som företaget hanterar. Dessa går man vidare och strukturerar i en konceptuell resursmodell (eng. conceptual model).

4 Steget från verksamhetsbeskrivning till systemkrav

Vid astrakan-modellering använder man sig av informationsutnyttjandet i processen för att identifiera kandidater till användningsfall, och rollerna i processen blir kandidater till (system)aktörer.

Vid rup-modellering använder man sig av den konceptuella resursmodellen där man använder verksamhetens roller som kandidater till (system)aktörer och namnger deras respektive användningsfall utifrån verksamhetsanvändningsfallen.

5 Systemutveckling med användningsfall och UML

Väljer man att driva sitt systemutvecklingsprojekt med hjälp av UML och användningsfall så bör det finnas en röd tråd från krav till testad kod enligt nedan.

Kravarbetet börjar med en visionsformulering utifrån resultatet av verksamhetsmodelleringen. Visionen bör i princip definiera det problem man försöker lösa med hjälp av ett IT-stöd, alternativt den möjlighet man vill ta tillvara på. Utifrån visionen identifierar man systemets aktörer och de användningsfall de är intresserade av.

Utfärdare
Andreas Hägglund
Godkännes

Datum
2009-11-21 17:51:00
Dokumentbeteckning

Utgåva

För varje användningsfall (eng. use case) gör man en användningsfallsbeskrivning (eng. use case specification), t.ex. i form av ett aktivitetsdiagram. Dessa utgör, tillsammans med övriga krav (kvalitetsattribut, informationsmodell m.m.) själva kravspecifikationen.

Efter kravarbetet vidtar analys och design-arbetet. Vi går nu in i lösningsarbetet, från vad till hur... För varje användningsfall gör vi nu en användningsfallsrealisering (eng. use case realization). I dessa beskriver man hur man går tillväga för att åstadkomma (realisera) den funktionalitet som är beskriven i användningsfallet. Ofta gör man detta med hjälp av ett antal olika diagram, såsom t.ex. sekvens-, klass-, objekt- och tillståndsdigram. Normalt använder man sig av minst ett dynamiskt perspektiv (t.ex. sekvensdiagram) och ett statiskt (t.ex. klassdiagram).

När man har realiserat ett tillräckligt antal användningsfall har man skapat x antal klassdiagram, y antal sekvensdiagram och z antal tillståndsdigram etc. Utifrån dessa skapar man analysmodellen. Grovt förenklat kan man säga att man slår samman alla olika klassdiagramen till ett, samtidigt som man behåller alla dynamiska diagram. Analysmodellen utgörs alltså av summan av realiseringarna.

Analysmodellen är lösningsoberoende. Det innebär att den beskriver en lösning som inte tar hänsyn till programspråk, operativsystem m.m. Designarbetet handlar bl.a. om att optimera analysmodellen för en viss teknik, t.ex. java. Arbetsmomenten är dock desamma som under analysarbetet. Dvs man utgår ifrån respektive analysdiagram och förfinar det ytterligare (till en designmodell).

Källkoden skrivs sedan utifrån designmodellen, och kompileras sedan till körbar kod. Denna kod systemtestas sedan med bl.a. hjälp av användningsfall och integrationstestas bl.a. utifrån sekvensdiagrammen i designmodellen.

6 Svårigheter

Ovanstående beskrivning är något förenklad och ger framför allt en teoretisk bild av ett väldigt praktiskt arbete.

I verkligheten så realiserar man sällan alla användningsfallen utan enbart de där man finner ett behov. Vad som utgör ett behov varierar från projekt till projekt. Man använder heller inte alltid samma diagram i respektive realisering utan även detta styrs av

Utfärdare
Andreas Hägglund
Godkännes

Datum
2009-11-21 17:51:00
Dokumentbeteckning

Utgåva

projektets behov och varierar från projekt till projekt. Slutligen är det inte ens säkert att man alltid väljer att göra en analysmodell. Ibland går man direkt ned till designmodellen och i de fall man faktiskt väljer att göra en analysmodell så är det inte alltid så att man alltid väljer att behålla och underhålla den. Ibland låter man analysmodellen helt enkelt övergå i en designmodell.

7 Om författaren

Andreas Hägglund arbetar som konsult på Systemvaruhuset, med verksamhetsutveckling, kravanalys och metodik som huvudinriktning. Han fungerar dessutom som lärare åt Astrakan Strategisk utveckling i kurser med inriktning på krav, verksamhetsmodellering, UML och RUP.

8 Mer information

Business modeling with UML, Penker & Eriksson, ISBN 0471295515

The object advantage, Jacobsson, ISBN 0201422891

www.systemvaruhuset.se