Actor: noget der har samme opførelse, som et mennesker f.eks. en ekspedient

Scenario: er når der sker en bestemt sekvens af aktioner og interaktion mellem actor og systemet ( det bliver også kaldt ”use case instance”)

Main Scenario: hvordan systemet normalt bliver brugt

Alternate Scenario: hvis der sker noget andet i systemet, som der ikke sker normalt

SuD: system under discussion

Primary actor: bruger mål er ud fuldt igennem SuD metoden f.eks. en ekspedient

Supporting actor: giver en service f.eks. information (ofte er det et computer system)

Offstage actor: nogen der har en interesse for casen, men som ikke er Primary actor eller Supporting actor. (f.eks. staten)

Brief use case: et afsnit der opridser hoved Scenarie

Casual use case: flere afsnit, der forklar flere forskellige Scenarie

Fully dressed use case: alle trinene for de forskellige Scenarier i detalje, og der er afsnit med forudsætning og succes garantier

Scope: et scope binder systemet under designet, i denne situation bliver det også kaldt for system use case. Scope kan også forklare hvordan en forretningen er brugt, af ens forbruger, sådan en proces bliver kaldt business use case

User-goal level: bliver ofte brugt til at beskrive, hvordan man opnår at opfylde den primær actor. Det svare rundt regnet til EBP.

EBP: elementary business process

Subfunction-level: bliver ofte brugt til af forklar de mindre trin til at nå bruger målet, og ofte bliver brugt til at man ikke gentager det man har gjort

Preconditions: udmeldinger, som må altid være sande før Scenariet er begyndt på use case.

Success guarantees( or postconditions): udmeldinger, som må være sande, efter færdiggørelse efter user casen eller efter færdiggørelsen af main Scenariet eller en alternativ vej.

Essential style writing: ”hole interfacet ud, og fokuser på hensigt

Concrete use case style: bruger interface afgørelser, til at indlejre ind i en use case tekst.

Black-box use cases: er den mest almenlige og anbefalede slags. De beskriver ikke indre arbejde for systemet, et komponenter eller design. Men beskriver at have ansvar, som er den mest almenlige samlede metaforiske tema i object-oriented thinking. (6.13)

Hvis man laver et godt case diagram, for man et godt ”context diagram”

Domian rules: som også bliver kaldt for forretnings regler

SS: Supplementary Specification  
  
Use-case driven deveopment: at man bruger de primære use-case modeler (6.21)