

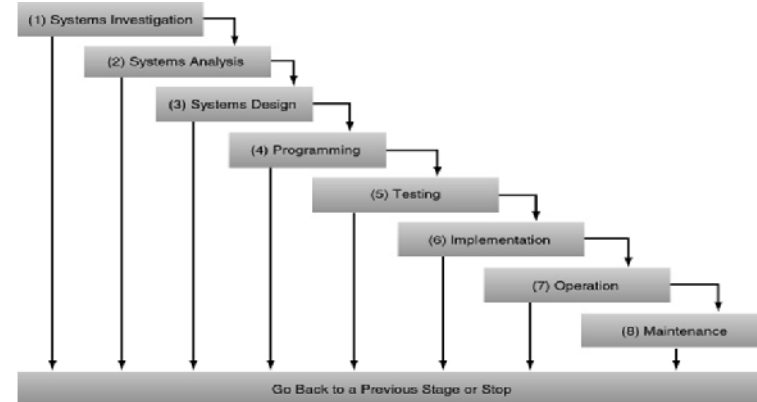
MSF procesni model

- *Framework* - podrška metodologijama
- MSF discipline
- MSF timski model
- Kako organizovati projektne timove
- Kako upravljati kompromisima (*tradeoffs*)
- Faze u MSF procesnom modelu

Zašto modeliranje?

- Zašto je modeliranje važno za razumevanje poslovnih informacionih sistema?
- Ne mogu se razumeti poslovni IS bez modela koji predstavljaju sisteme u njihovoj realnosti.
- Za ve inu ljudi, sistemi su preveliki i kompleksni da bi se shvatili u celosti (npr. inženjeri koji prave automobile kreiraju modele pre nego što naprave nova kola; arhitekta prave modele zgrada koje projektuju itd.)

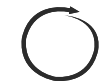
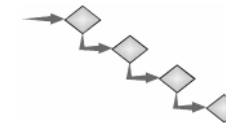
Životni ciklus razvoja sistema (System Development Life Cycle - SDLC)



- SDLC - strukturirani okvir koji se koristi za velike IT projekte koji se sastoje od sekvencionalnih procesa po kojima se razvijaju informacioni sistemi.

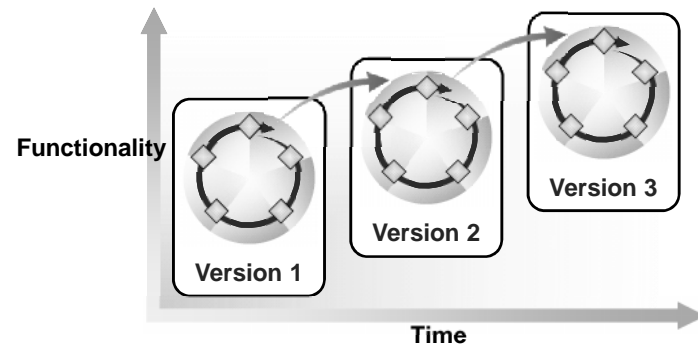
Microsoft Solution Framework - MSF

- MSF obezbeđuje skup modela, pravila i smernica za projektovanje i razvoj poslovnih rešenja na način koji osigurava da se svim elementima projekta (ljudi, procesi, alati) uspešno upravlja.
- Obezbeđuje dva različita pristupa životnog ciklusa projekta: model vodopada i spiralni model.
 - Model vodopada* – svi zadaci u jednoj fazi se moraju završiti pre nego što se pređe na sledeću fazu.
- Spiralni model* – model se bazira na kontinualnoj potrebi za usavršavanjem zahteva. Ovaj model je pogodan za brzi razvoj aplikacija (*Rapid Application Development – RAD*).



Kako MSF procesni model radi

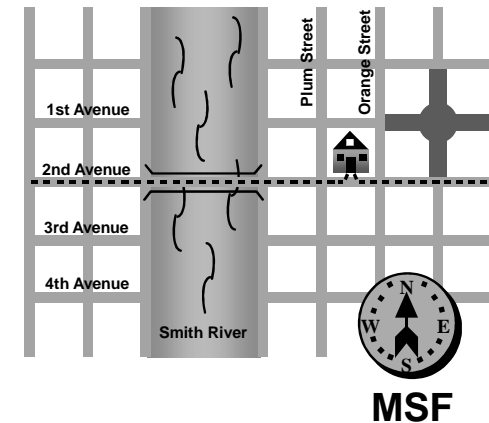
- MSF procesni model kombinuje najbolje principe modela vodopada i spiralnog modela.
- Spiralnost se definiše i u okviru samog projekta, a i u okviru kritičnih tačaka (*milestones*).
- Microsoft-ov primer verzionisanja istog proizvoda:



Framework podrška metodologijama

Metodologija
daje tačne smernice
za poznato odredište

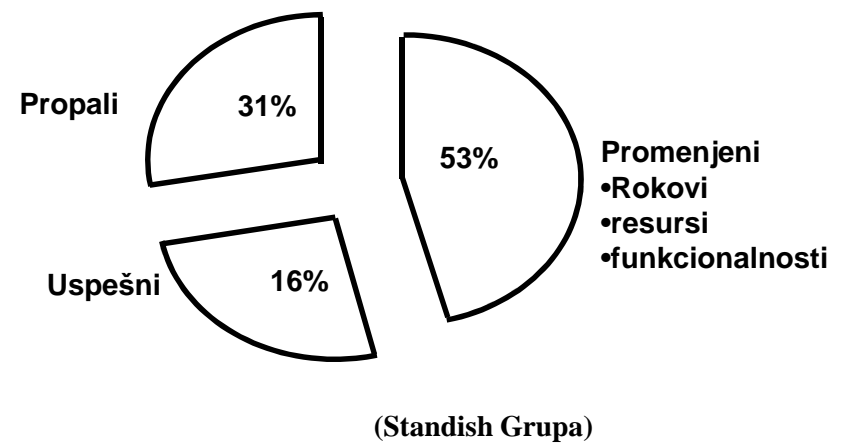
Framework
kao kompas,
potvrđuje
napredovanje i daje
upravljačke smernice



Zašto MSF?

- 16 % uspešnih projekata
- 31% neuspešnih projekata
- 53% projekata koji su uspeli, ali su probili rokove ili su isporu eni sa lošijom funkcionalnoš u i dr.
- Sa MSF-om zabeležen je rast od 20% uspešnih projekata sa svakodnevnom tendencijom porasta.

Zašto MSF?



MSF discipline

Upravljanje
timom

MSF

Upravljanje
rizicima

Upravljanje
procesima

MSF discipline

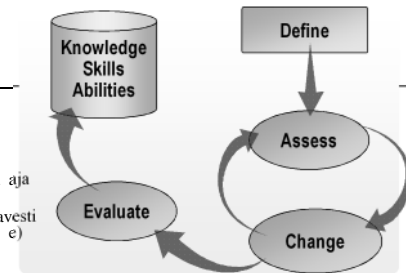
□ MSF uključuje discipline za upravljanje ljudima, procesima i tehnologijom:

1. Upravljanje rizicima (*Risk management*):

- identifikovanje rizika (*brainstorming*)
- analiza rizika (sortirati ih, proceniti verovatno u dogajajućim rizicima)
- planiranje rizika (proceniti uticaj na ishod projekta, navesti na koje rizike kako da se smanje, šta uraditi ako do rizika dođe)
- praćenje rizika
- kontrolisanje rizika
- uklanjanje rizika

2. Spremnost (*Readiness management*) – kako podizati nivo spremnosti cele ekipe (motivacija, edukacija, interesantnost, iskustvo i dr.):

- Definisanje scenarija, kompetencija i nivoa spremnosti neophodnih za uspešno planiranje, kreiranje i upravljanje rešenjem.
- Procena – analiza kompetencija u odnosu na različite uloge u poslu.
- Promene – poboljšanje veština kroz obuke i praćenje napredovanja.
- Ocena – tim određuje da li su planovi obuke bili efektivni i da li je potrebno znanje uspešno implementirano na poslu.



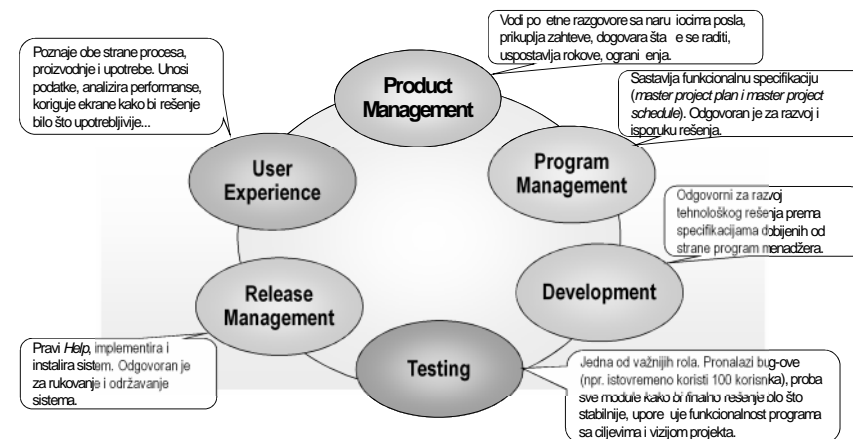
3. Upravljanje projektima (*Project management*):

- Mnoge funkcije projekta menadžera je preuzela program menadžment rola.
- Definiše i upravlja ciljem i oblašću projekta.
- Integriše planiranje i upravljanje promenama.
- Priprema budžet i upravlja troškovima.
- Priprema i prati raspored odvijanja projekta.
- Osigurava da su odgovarajući resursi alocirani na projektu.
- Olakšava timsku i spoljnu komunikaciju.
- Olakšava proces upravljanja rizicima.
- Dokumentuje i nadgleda procesom upravljanja kvalitetom tima.

MSF timski model

- MSF timski model specificira šest različitih uloga i ističe važnost jasnih uloga, odgovornosti i ciljeva članova.
- Fleksibilnost MSF timskog modela omogućava da se brže prilagodite ciljevima projekta, veličini tima i vještinama članova tima.
- Ne postoji uloga projekt menadžera, niti rigidna hijerarhijska struktura u procesu odlučivanja. Sve uloge u timu ispunjavaju određeni cilj, dok se glavne odluke donose konsenzusom članovog tima. Ukoliko se konsenzus ne može postići, program menadžment rola donosi finalnu odluku.

Kako organizovati projektne timove



Kako upravljati kompromisima (*tradeoffs*)

□ Da bi uspešno definisali cilj i upravljali projektom, neophodno je:

- identifikovati ograničenja projekta
- upravljati kompromisima
- uspostaviti kontrolu promena
- pratiti napredak projekta.



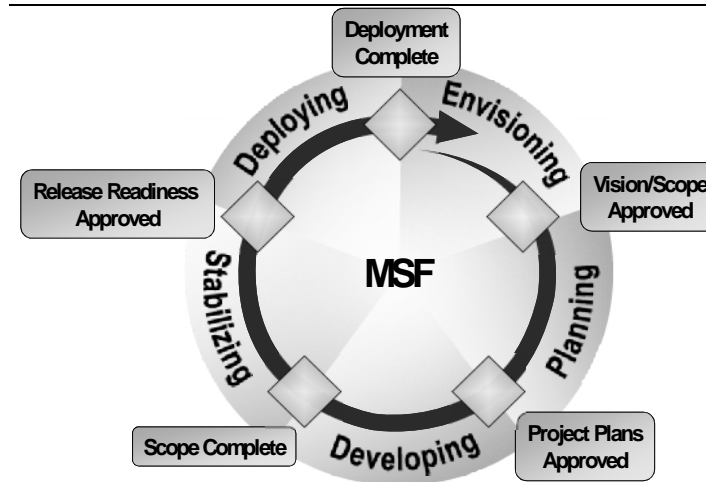
a) Tradeoff triangle

	Fiksirani	Odobrani	Prilagodljivi
Resursi	✓		
Raspored		✓	
Funkcionalnost			✓

b) Tradeoff matrix

- U projektima postoji jasan odnos između resursa, rasporeda i funkcionalnosti projekta.
- S obzirom da je skoro nemoguće ostvariti istovremeno sve ciljeve, neophodno je upravljati kompromisima.

Faze u MSF procesnom modelu



Prikupljanje i analiziranje informacija

- Upotreba notacija modeliranja
- Kreiranje sluajeva koriš enja (*use cases*) i scenarija (*usage scenarios*)
- Prikupljanje informacija
- Analiziranje informacija

Koristi od modeliranja

- Modeliranje može pojasniti prezentaciju složenih problema.
- Modeliranje trenutnog stanja identifikuje:
 - zahteve
 - probleme i rizike
 - izgubljene informacije
- Dve naj eš e koriš ene notacije su:
 - Unified Modeling Language (UML) i
 - Object Role Modeling (ORM).

Šta je UML?

- UML je jezik za vizuelizaciju i modeliranje softverskih sistema.
- UML se može koristiti za:
 - kreiranje specifikacija
 - izgradnju modela
 - dokumentovanje modela

UML dijagrami

- Svaki tip dijagrama pokazuje različite poglede:
 - **dijagram klase (Class)** – opisuje različite klase i njihova spajanja (*associations*).
 - **objektni dijagram (Object)** – opisuje različite objekte u sistemu i njihove međusobne veze.
 - **dijagram slučaja (Use case)** – prikazuje željenu funkcionalnost sistema.
 - **dijagram komponenti (Component)** – prikazuje implementacioni pogled na sistem odnosno prikazuje različite komponente sistema i njihove veze (npr. *source code*, *object code* i *execution code*).
 - **dijagram implementacije (Deployment)** – predstavlja spajanje softverskih komponenti sa vorovima fizike implementacije sistema.
 - **dijagram saradnje (Collaboration)** – predstavlja skup klasa i njihovih poslanih i prijimljenih poruka.
 - **dijagram sekvenci (Sequence)** – opisuje interakciju između klasa koja predstavlja redosled poruka koje se razmenjuju između klasa.
 - **dijagram stanja (State)** – je algoritam različitih stanja jednog objekta koji se događaju pri uticaju različitih spoljnih događaja.
 - **dijagram aktivnosti (Activity)** – predstavlja drugi način pogleda na stanje i uključuje i sekvence aktivnosti. Pogodan kod modeliranja konkurentnih procesa kako bi se videla neophodna sinhronizacija između klasa i zadataka.

ORM

- ORM je metodologija za modeliranje podataka koji se odnose na poslovne zahteve.
- ORM se koristi za dokumentovanje poslovnih pravila i projektovanje baza podataka, a finalni rezultata je model baze podataka.
- ORM koristi prirodan jezik i intuitivne dijagrame.
- Informacije se predstavljaju kao elementarne injenice koje pokazuju da objekti imaju neka svojstva ili da su jedan ili više objekata u vezi.

Slučajevi korišćenja (*use cases*)

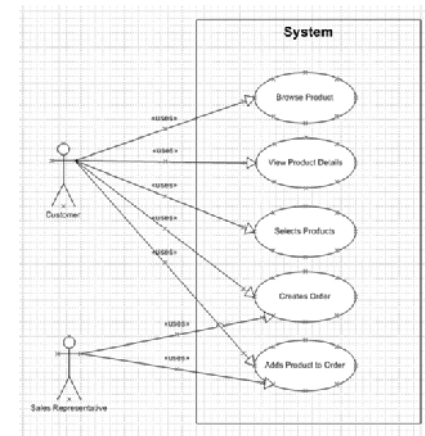
- *Use case*-ovi opisuju učesnike (*actor*), objekte (*objects*) i akcije (*actions*) koji dostižu ciljeve u sistemu.
- *Use cases* identifikuju:
 - granice sistema
 - sekvencu zadataka i hijerarhije
- Na primer:
 - *Prodavac pravi porudžbinu*
 - *Menadžer prodaje odobrava vešću i popust*

Use case iskazi u spreadsheet-u

	A	B	C	D	E	F	G
1	Use Case Examples						
2							
3	Sales Representative creates order						
4	Sales Representative adds product to order						
5	Customer browses product						
6	Customer views product details						
7	Customer selects product						
8	Customer creates order						
9	Customer adds product to order						
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							

UML use-case dijagram

- UML dijagram koji predstavlja prethodne *use case* iskaze



Kako izvući i use case-ove

- Pronađi i zadatak analiziraju i izvore informacija

Izvod iz intervjua: “Da bi identifikovali naše najbolje kupce i razloge zbog čega su oni najbolji, prodajno osoblje treba da pronađe način da pristupi i analizira našim podacima o prodaji”

- Da bi napravili *use case* dijagram, neophodno je identifikovati sledeće:
 - sistem
 - učesnike – entitet koji je u interakciji sa sistemom radi izvršenja nekog događaja. Mogu biti korisnici sistema ili entiteti kao što su neki drugi sistem ili baza podataka koja je smeštena van sistema.
 - veze između učesnika i sistema – samo one veze koje su bitne za poslovanje
 - granice sistema

Kako izvući i use case-ove

- Za svaki zadatak odrediti:
 - Ko izvršava akciju - *actor*
 - Akciju
 - Objekat akcije

<i>Actor</i>	<i>Action</i>	<i>Object</i>
Prodajno osoblje	pristupa	podacima o prodaji
prodajno osoblje	analizira	podatke o prodaji
prodajno osoblje	identifikuje	najbolje klijente

Šta su scenarija upotrebe (Usage Scenarios)?

- *Usage scenarios* opisuje šta *use case* radi
 - *use case* opisuje visok nivo interakcije između učesnika i sistema, s toga jedan *use case* obično zahteva nekoliko scenarija kako bi se detaljno opisao
 - *usage scenarios* obezbeđuju dodatne informacije o sekvencama aktivnosti i zadataka od kojih se sastoji jedan proces.
 - *usage* scenarija dokumentuju sekvence zadataka.
- *Usage scenarios* opisuju izuzetke
 - Izuzeci su atipični događaji ili alternativne sekvence zadataka.

Kako se kreira Usage Scenario

1. Odrediti preduslove
 - Šta su neophodni preduslovi da bi se krenulo u izvršenje *usage* scenarija?
2. Identifikovati posledice
 - Šta je posledica uspešno obavljene akcije?
3. Razbiti aktivnosti na korake
 - Raspored svih poslova odnosno svakog pojedinačnog posla, kako bi posao bio obavljen.
4. Identifikovati izuzetke koji se mogu dogoditi na svakom koraku.
 - Možda će biti neophodno da se naprave *usage* scenarija za ove izuzetke
5. Identifikovati zahteve na koje *usage scenario* ukazuje
6. Identifikovati izvor za *usage scenario*, odnosno na koji se *use case* odnosi

Primer: Uplata telefonskog računa

- Preduslovi
 - ako nema struje transakcija neće moći da se izvrši
- Posledice izvršenja *usage* scenarija
 - evidencija plaćenog računa
- Razbijanje na pojedinačne poslove
 - službenik vrši naplatu telefonskog računa
- Izuzetak – skretanje na drugi *usage scenario* ili obaranje transakcije
 - ukoliko nema dovoljno novca za plaćanje, onda prelazimo na drugi *usage scenario*.

Zašto kreirati *usage scenario* trenutnog stanja?

- Trenutno stanje scenarija (*current state usage scenarios*) opisuje kako se poslovne aktivnosti trenutno odvijaju odnosno kako trenutno proces izgleda.
- Buduće stanje scenarija (*future state scenario*) predstavlja željene buduće aktivnosti odnosno kako bi *usage scenario* trebao da izgleda.
- *Current state usage scenarios* vam pomaže da:
 - identifikujete probleme u sistemu
 - odredite ciljeve
 - otkrijete nepodudarnosti između percepcije trenutnog problema i stvarnog problema

Kako identifikovati zahteve

- Kreirati listu zahteva tokom procesa prikupljanja informacija i identifikovati izvor odakle je zahtev došao.
- Zahtevi moraju da budu kratke (ostvarljive) rešenice.
- Proširiti listu zahteva
 - preispitati sve prikupljene informacije
 - pronaći zahteve
- Zahteve podeliti od želja
- Prepoznati ograničenja i pretpostavke
 - Ograničenje je fiksni limit, kao npr. budžet
 - Razjasniti pretpostavke kako bi se zaštitili od nesporazumevanja
- Identifikovati skrivene zahteve
 - Neki zahtevi nisu odmah vidljivi (npr., zakonske regulative koje se menjaju).

Prikupljanje informacija

- Kategorije informacija:
 - *Poslovne* – opisuje kako se radi posao, odnosno opisuje funkcije i unakrsne funkcije koje posao obavlja. Informacije iz ove kategorije opisuju i primarne ciljeve, proizvode i usluge, finansije, organizacionu strukturu.
 - *Aplikacije* – automatizovane i neautomatizovane usluge koje podržavaju poslovne procese. Uključuju gotove aplikacije, komponente, izvorne kodove koje su dostupne za analizu informacija ili funkcionalnosti zadatka.
 - *Operacije* – informacije koje su neophodne da bi se pokrenuo poslovni proces.
 - *Tehnologije* – tehničke usluge neophodne za izvršenje i podršku poslovne misije. Uključuju topologije, razvojna okruženja, bezbednost, usluge mreže, DBMS, tehničke specifikacije, operativne sisteme, hardver i dr.

Tehnike za prikupljanje informacija

Tehnike	Opis
Praćenje (<i>shadowing</i>)	Direktno posmatranje kako pojedinac obavlja svoj posao kako bi se uvidela praksa i problemi.
Intervjui	Prikupljanje određenih informacija od pojedinaca. Nedostatak je što oni daju svoje viđenje posla i problema i ukoliko nije ekspert u domenu koji proučavamo.
Fokus grupe	Sa većom grupom se radi intervju kako bi se dobila prethodna informacija, otkrila ponašanja i deljena mišljenja.
Istraživanja	Prikupljanje detaljnih i statističkih podataka.
Instrukcije korisnika	Radi se njegov posao. Krajnji korisnik vas podučava kako oni rade sa sistemom.
Pravljenje prototipa	Simulirati sistem koji nije moguće testirati direktno. Pravi se prototip koji se usavršava.
Instrumented versions	Koriste se metode praćenja da bi se snimilo kako korisnik izvršava zadatke.

Analiziranje informacija

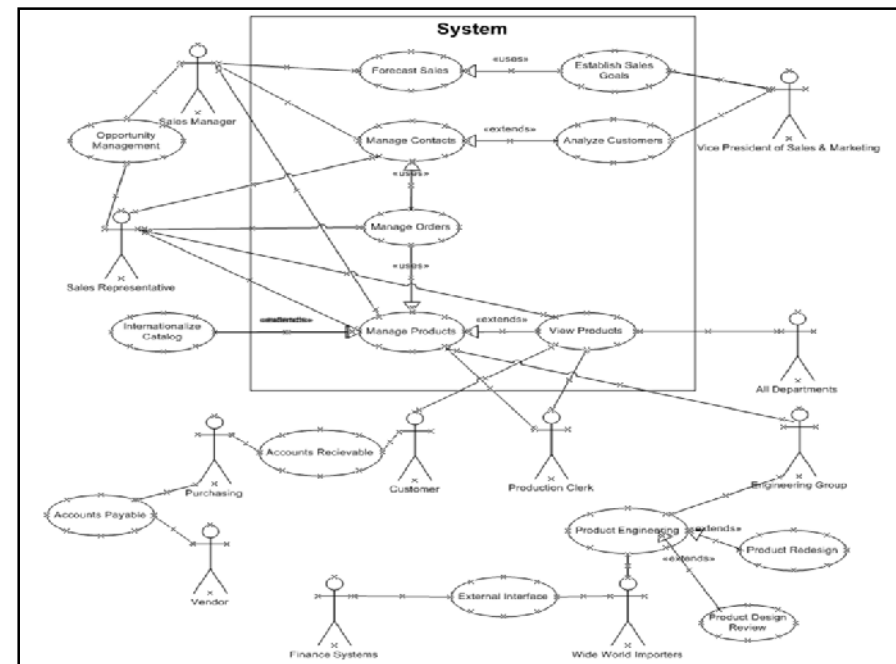
- Kada se prikupi dovoljno informacija, sledeći korak je filtriranje informacija koje su relevantne za poslovanje.
- Verifikovati da postoji dovoljno informacija koje govore o poslovnim zahtevima i zahtevima za rešenjem:
 - Potrebe sigurnosti
 - Podrška za rešenja i njihove karakteristike
 - Planirane promene u poslovanju koje mogu da utiču na projektovano rešenje
 - Performanse koje korisnici očekuju ili koje je neophodno da bi se održala konkurentnost
 - Integracija aplikacija – postojeće aplikacije koje treba da se povežu sa novim rešenjem

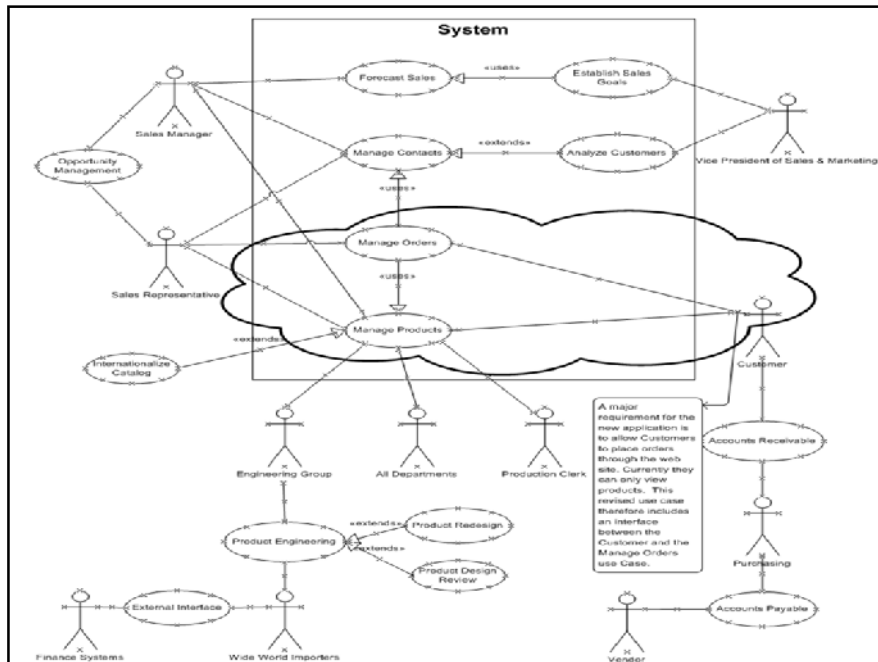
Po etak dokumentovanja zahteva

- Ve tokom intervjuja napisati listu potencijalnih zahteva
 - **Odvojiti zahteve od želja**
 - Grubi dokument (draft) o zahtevima izvu enih iz intervjuja i use case dijagrama

Orig ID	Opis zahteva	Prioriteti?	Izvor	Pitanja
1	Can synchronize with our online applications to record all the information collected throughout the day		Territory sales manager	
2	Retrieve the customer data and allow us to view the data in various ways	Y	Territory sales manager	
3	Find out who my best customers are		Territory sales manager	

(Na slede im slajdovima se prikazuje inicijalni use case dijagram koji predstavlja po etno stanje odnosno analizu teku eg stanja sistema i revidirani (revised) use case dijagram koji predstavlja onako kako bi trebalo da bude sa reinženjeringom procesa u sistemu.)





Interna dokumentacija projektnog tima

Dokumentacija	Identifikuje	Upotreba
Katalog u esnika (<i>actors catalog</i>)	Svakog u esnika po imenu	Lista odgovornosti za svakog u esnika i ukazivanje na daljne informacije
Katalog poslovnih pravila (<i>business rules catalog</i>)	Svako poslovno pravilo po ID broju, nazivu, opisu i autoritetu	Opisuje trenutne funkcionalnosti
Rezik (glossary)	Terminologija i definicije koje koriste razliiti stakeholders-i, poslovni procesi i projektni termini	Osigurava se konzistencija iste terminologije radi jasnijeg sporazumevanja