

Nxjerrja dhe analiza e kërkesave

Kërkesa është një vecori që një sistem duhet të ketë ose një kufizim të cilit sistemi duhet t'i nënshtrohet në mënyrë që të pranohet nga klienti. Inxhinierimi i kërkesave ka si qëllim përcaktimin e kërkesave të sistemit që do të ndërtohet. Nxjerrja e kërkesave dhe analiza janë dy faza të rëndësishme të inxhinierimit të kërkesave. Rezultati i nxjerrjes së kërkesave është specifikimi i sistemit në nivel të tillë që ta kuptojë klienti. Analiza ka si rezultat modelin e analizës i cili mund të interpretohet në mënyrë të qartë nga zhvilluesit. Nxjerrja e kërkesave është një aktivitet me i ndërlikuar sesa analiza, sepse kërkon bashkëpunimin e disa grupeve pjesëmarrëse me formime të ndryshme. Klientet dhe përdoruesit janë përgjithësisht ekspertet e fushës së aplikimit dhe kanë një ide të përgjithshme të asaj që sistemi duhet të bëjë, por ata kanë pak përvojë përse i perket zhvillimit të software. Nga ana tjetër, zhvilluesit kanë përvojë në zhvillime sistemesh, por kanë pak njohuri për mjedisin ku përdoruesit veprojnë çdo ditë.

Mënyra për të kapërcyer këtë boshllëk ndërmjet përdoruesve dhe njerezve teknike është përdorimi i skenareve dhe rasteve të përdorimit. Një *skenar* përshkruan një shembull të përdorimit të sistemit duke përshkruar një sekuencë ndërvprimesh ndërmjet përdoruesit dhe sistemit. Një *rast përdorimi* është një abstragim që përshkruan një klasë skenarësh. Skenaret dhe rastet e përdorimit shkruhen shpesh në gjuhë natyrore, një formë e kuptueshme nga përdoruesi.

1. Nxjerrja e kërkesave (Requirements Elicitation)

Nxjerrja e kërkesave fokusohet tek përshkrimi i qëllimit të sistemit. Klienti, zhvilluesit dhe përdoruesit identifikojnë një problem dhe përcaktojnë një sistem që ndihmon në zgjidhjen e problemit. Ky përcaktim quhet **specifikimi i kërkesave** dhe shërben si kontrata ndërmjet klientit dhe zhvilluesve. Specifikimi i kërkesave strukturohet dhe formalizohet gjatë analizës duke prodhuar **modelin e analizës**. *Specifikimi i kërkesave dhe modeli i analizës prodhojnë të njëjtën informacion, por ndryshojnë në gjuhën dhe simbolikën që përdorin për ta përshkruar informacionin*. Specifikimi i kërkesave shkruhet në gjuhë natyrore, ndërsa modeli i analizës shprehet me simbolike formale ose gjysëm formale. Specifikimi i kërkesave mbështet komunikimin me klientin dhe përdoruesin. Modeli i analizës mbështet komunikimin mes zhvilluesve. Të dyja janë modele të sistemit në kuptimin që përpiqen të përfaqësojnë në mënyrë të saktë aspektet e jashtme të sistemit. Mëqenëse të dy modelet përfaqësojnë të njëjtat aspekte të sistemit, nxjerrja e kërkesave dhe modelimi i analizës ndodhin njëkohësisht dhe në mënyrë iterative.

Nxjerrja e kërkesave dhe analiza fokusohen vetëm tek ajo që sheh përdoruesi nga sistemi. P.sh, funksionalitetet e sistemit, ndërvprimi mes përdoruesit dhe sistemit, gabimet që sistemi duhet të gjejë dhe të menaxhojë, kushtet e mjedisit në të cilin sistemi funksionon, janë pjesë e kërkesave. Struktura e sistemit, teknologjia e implementimit për sistemin, modelimi i sistemit, metodologjia e zhvillimit dhe aspekte të tjera që nuk lidhen drejtpërdrejtë me ato që është e dukshme për përdoruesin, nuk janë pjesë e kërkesave.

Aktivitetet kryesore të nxjerrjes së kërkesave janë:

- *Identifikimi i aktoreve*: identifikohen tipet e ndryshme të përdoruesve të sistemit të ardhshëm.

- *Identifikimi i skenareve*: zhvilluesit vëzhgojnë përdoruesit dhe krijojnë skenare të detajuara të funksionaliteteve tipike që do të ketë sistemi i ardhshëm. Skenaret janë shembuj konkretë të sistemit. Zhvilluesit përdorin këto skenare për të komunikuar me përdoruesin dhe për të thelluar të kuptuarit e fushës së aplikimit.
- *Identifikimi i rasteve të përdorimit*: kur zhvilluesit dhe përdoruesit bien dakort mbi bashkësinë e skenareve, zhvilluesit derivojnë nga këto skenare një bashkësi me raste përdorimi që përfaqësojnë plotësisht sistemin e ardhshëm. Skenaret janë shembuj konkretë që ilustronjë një rast përdorimi të vetëm, ndërsa një rast përdorimi është një abstragim që përshkruan të gjitha rastet/ skenaret e mundshme.
- *Permiresimi i rasteve të përdorimit*: zhvilluesit sigurohen që specifikimi i kërkesave është i plotë, duke detajuar çdo rast përdorimi dhe udhë përshkruar sjelljen e sistemit në praninë e gabimeve dhe rasteve të vecanta.
- *Identifikimi i lidhjeve mes rasteve të përdorimit*: zhvilluesit identifikojnë varesitë ndërmjet rasteve të përdorimit. Konsolidohet modeli i rasteve të përdorimit duke faktorizuar sjelljet e përbashkëta. Kjo siguron konsistencë në specifikimin e rasteve të përdorimit.
- *Identifikimi i kërkesave jofunksionale*: zhvilluesit, përdoruesit dhe klientët bien dakort për aspekte që janë të dukshme për klientin, por që nuk lidhen drejtpërdrejt me funksionalitetet. Këto përfshijnë kufizime në performancën e sistemit, dokumentim, burimet që kërkon, sigurinë dhe cilësinë.

Gjatë nxjerrjes së kërkesave, zhvilluesit aksesojnë burime të ndryshme informacioni si dokumenta të ofruara nga klientët, dokumenta për fushën (domainin) e aplikimit, manuale, dokumenta teknike të sistemeve aktuale që do të zëvendësohen nga sistemi i ri, dhe sigurisht informacionet e marra nga përdoruesit dhe klientët. *Ndërveprimi me i madh mes klienteve, përdoruesve dhe zhvilluesve ndodh gjatë nxjerrjes së kërkesave.*

2. Koncepte të nxjerrjes së kërkesave

2.1 Kërkesat Funksionale

Kërkesat funksionale përshkruajnë ndërveprimet mes sistemit dhe mjedisit pavarësisht nga menyrat e implementimit të tij. Mjedişi përfshin përdoruesin dhe çdo sistem tjetër të jashtëm me të cilin do të komunikojë sistemi i ardhshëm. Shembulli i kërkesave funksionale për *SatWatch*, një orë që sinkronizohet pa pasur nevojën e ndërhyrjes nga jashtë të përdoruesit do të ishte si më poshtë:

SatWatch është një orë dore që tregon orën bazuar në vendodhjen e momentit. Ora përdor satelitet GPS (Global Positioning System) për të përcaktuar vendodhjen dhe, strukturat e brendshme të të dhënave për të dedektuar zonën kohore bazuar në vendodhjen aktuale.

Informacioni i ruajtur në *SatWatch* dhe saktësia e matjes së kohës janë të tilla që përdoruesi nuk ka nevojë të rregullojë orën. *SatWatch* rregullon orën dhe datën ndërkohe që përdoruesi kalon nga një zonë kohore në tjetrën ose kur ai kalon kufinjtë shtetërore. Për këtë arsye, *SatWatch* nuk ka butonë që t'i ofrojnë mundësi përdoruesit që ai të ndërshoje orën.

SatWatch e percakton vendodhjen duke perdorur satelitet GPS dhe, si e tille, i nenshtrohet te njejtave kufizimeve qe i nenshtrohet dhe sistemi GPS (psh. Paaftesia per te percaktuar vendodhjen ne moment te caktuara te dites ne zonat malore). Gjate periudhave te shkeputjes me satelitet, SatWatch supozon qe nuk ndodhin kalime nga nje zone ne tjetren ose nga nje zone kufitare ne tjetren. SatWatch e rregullon oren sapo lidhet me sistemin.

SatWatch ka dy ekrane per shfaqjen e ores, minutave, sekondave ne njerin dhe dites, dates, muajit dhe vitit ne tjetrin. Teknologjia e ekraneve duhet te jete e tille qe te lejoje leximin e informacionit dhe ne drite te dobet.

Ne rastin e kalimit te kufinjve shteterore, perdoruesi i ores mund te ndryshoje programin e ores duke perdorur paisjen WebifyWatch (qe jepet bashke me oren) dhe nje kompjuter te lidhur me Internetin.

Kerkesat funksionale me lart perqendrohen vetem tek nderveprimet e mundshme ndermjet ores dhe botes se jashtme (perdoruesit te ores, GPS, WebifyWatch). Pershkrimi i mesiperm nuk perqendrohet tek detajet e implementimit (psh. procesori, gjuha, teknologjia e ekranit shfaqes).

2.2 Kerkesat jo funksionale

Kerkesat jo funksionale pershkruajne aspekte te sitemit qe nuk lidhen drejtperdrejt me sjelljen funksionale te sitemit. Kerkesat jo funksionale perfshijne kerkesa qe lidhen me aspekte te ndryshme te sitemit duke filluar nga perdorueshmeria (usability) tek performanca. Tek kerkesat jo funksionale mund te permenden:

- **Perdorueshmeria (usability):** lehtesia me te cilin perdoruesi mund te mesoje te veproje me sistemin, te pergatitet per inpute dhe te interpretojte rezultatet e sitemit. Shembuj jane konvencionet per nderfaqen grafike, hapesira e ndihmes/tutorialeve online, niveli i dokumentimit per perdoruesin. Shpesh, klientet i adresojne ceshtjet e perdorueshmerise duke i kerkuar zhvilluesve qe te ndjekin nje linje te caktuar ne perdorimin e ngjyrave, logove, fonteve.
- **Besueshmeria(reliability):** aftesia e sitemit apo komponentes per te kryer funksionet e kerkuara ne kushte te caktuara per nje periudhe kohe. Shembuj jane: koha mesatare e pranueshme nga momenti kur ndodh nje problem deri ne detektimin e ketyre gabimeve apo dallimin e sulmeve te sigurise. Kohet e fundit kjo kategori eshte zevendesuar me **varesine (dependability)**, qe eshte ajo karakteristike e sitemit qe ben te mundur qe sitemit te mund t'i besohet ne menyre te justifikueshme per sherbimet qe ai ofron. Varesia perfshin besueshmerine, **fortesine (robustness)** qe eshte shkalla ne te cilin nje sistem ose komponente mund te funksionojte ne menyre korrekte ne kushtet e pranise se inputeve jo te sakte ose kushteve me ngarkese te madhe ne mjedis, **siguria (safety)** qe eshte nje mates i mungeses se pasojave katastrofike per mjedisin.
- **Performanca:** kerkesat e performances lidhen me attribute sasiore te sitemit si *koha e pergjigjes* (sa shpejt reagon sistemi ndaj inputit te perdoruesit), **raporti output/ input (throughput)** qe tregon sa pune mund te kryeje sistemi gjate nje intervali kohor te caktuar,

disponueshmeria (availability) qe eshte shkalla ne te cilen nje sistem apo komponent eshte operacional dhe i aksesueshem kur kerkohet qe te perdoret dhe, saktesia.

- **Mbeshtetja (supportability):** lehtesia e ndryshimeve te sistemit pasi ai eshte leshuar. Ketu perfshihen **pershtatshmeria (adaptability)**- aftesia per te ndryshuar sistemin qe t'u pergjigjet koncepteve te reja te domainit, **mirembajtja (maintainability)**- aftesia per te ndryshuar sistemin per te futur teknologjite e reja apo per te riparuar defekte, **portability** – lehtesia per te zhvendosur nje sistem apo komponente nga nje mjedis hardware ose software ne nje tjetër.

Kerkesa jo funksionale per SatWatch

- Cdo perdorues qe di te lexoje nje ekran dixhital dhe qe kupton shkurtime e zonave kohore internacionale duhet te jete ne gjendje ta perdore SatWatch pa nje manual perdoruesish [kerkese perdorshmerie].
- Mqs SatWatch nuk ka butona, nuk duhet te ndodhe asnje gabim ne software qe kane te bejne me rregullimin e kohes [kerkesa besueshmerie].
- SatWatch duhet te shfaqe zonen kohore te sakte Brenda 5 minutash nga perfundimi i nje peridudhe shkeputjeje me GPS [kerkesa performance].
- SatWatch duhet te mase oren brenda 1/100 e sekondes per 5 vjet [kerkesa performance].
- SatWatch duhet te shfaqe oren e sakte ne te gjitha 24 zonat kohore [kerkesa performance].
- SatWatch duhet te pranoje ndryshime (upgrades) nepermjet nderfaqes me Webify Watch [kerkesa mbeshtetjeje].

2.3 Plotesia, Konsistenca, Qartesia dhe Korrektesia

Kerkesat validohen vazhdimisht me klientin dhe perdoruesin. Validimi eshte nje hap kritik ne procesin e zhvillimit, mqs dhe klienti dhe zhvilluesi varen nga specifikimi i kerkesave. Validimi i kerkesave perfshin kontrollin qe specifikimi eshte i plote, konsistent, pa dykuptimesi dhe i sakte. Specifikimi eshte i **plote** nqs jane perfshire te gjitha skenaret e mundshem, duke perfshire sjelljet e vecanta. Specifikimi eshte **konsistent** nqs ai nuk eshte kontradiktor. Specifikimi eshte pa dykuptimesi nqs eshte specifikuar saktesisht vetem nje sistem (pra nuk ka vend per dy ose me shume interpretime). Specifikimi eshte korrekt nese ai perfaqeson ne menyre te sakte sistemin per te cilin ka nevojë klienti dhe qe zhvilluesit kane ndermend te ndertojne.

Kontrolle te bera tek specifikimi i kerkesave gjate validimit

I plote – te gjitha karakteristikat me interes jane te pershkruara tek kerkesat.

Kerkese jo e plote: Specifikimi i kerkesave te SatWatch nuk specifikon sjelljen kur perdoruesi eshte ne zonen e kufirit te sakties se GPS.

Zgjidhja: te shtohet nje kerkese funksionale qe shpreh kerkesen qe koha qe tregon SatWatch nuk duhet te ndryshoje me shpesh sesa cdo 5 minuta.

Konsistente – nuk ka kërkesa që janë kontradiktore me njëra tjetrën.

Kërkesa jo konsistente: Ora që nuk ka gabime në software nuk ka nevojë të ketë një mekanizëm që të shkarkojë versionet e reja të software

Zgjidhja: rishikohet njëra nga kërkesat që shkaktojnë konflikt në specifikim (psh. Shprehet në mënyrë tjetër kërkesa që ora të mos ketë gabime)

Dyktimesia - një kërkesë nuk mund të interpretohet në dy mënyra të ndryshme

Shembull: specifikimi i SatWatch i referohet zonës kohore dhe kufinjve shtetërorë. A e merr parasysh Satwatch ndryshimin e orës për shkak të kursimit të energjisë, apo jo?

Zgjidhje: shtohet një kërkesë që sqaron konceptin e ndryshimit të orës për shkak të ndryshimit të energjisë.

2.4 Realizmi, Verifikueshmeria, Gjurmimi

Specifikimi i kërkesave duhet të jetë:

realist- sistemi mund të implementohet brenda kufizimeve

i verifikueshem – kur ndërtohet sistemi, duhet të jetë e mundur të ndërtohen teste që demostrojnë që sistemi i përmbush kërkesat e specifikuara. Kërkesa të pa verifikueshme janë:

- *Produkti duhet të ketë një ndëfaqe grafike të mire.* – E mire nuk është e përcaktuar
- *Produkti duhet të jetë pa gabime.* – Kërkon sasi të mëdha burimesh
- *Produkti duhet t'i përgjigjet përdoruesit brenda 1 sekonde për pjesën më të madhe të rasteve.* – “Pjesa më e madhe e rasteve” është e papërcaktuar.

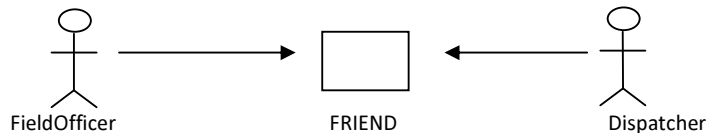
Specifikimi i kërkesave është i **gjurmueshem** nëse çdo kërkesë mund të gjurmohet përgjatë zhvillimit të software deri tek funksioni korrespondues i sistemit dhe, nëse çdo funksioni i gjendet kërkesa korresponduese. Gjurmimi ka të bëjë dhe me gjetjen e varesive ndërmjet kërkesave, funksioneve të sistemit, dhe produkteve të ndërmjetme që përfshijnë komponente, klasa, metoda, attribute objektësh. Gjurmimi është shumë i rëndësishëm në zhvillimin e testeve dhe për vlerësimin e ndryshimeve. Nëse kërkesat janë të gjurmueshme, testuesi është në gjendje të identifikojë nëse funksionaliteti janë testuar apo jo. Kur vlerësohen ndryshimet, gjurmimi bën të mundur që analisti dhe zhvilluesit të identifikojnë të gjithë komponentet dhe funksionalitetet që do të ndikohen nga ndryshimi.

2.5 Aktivitetet e nxjerrjes së kërkesave

2.5.1 Identifikimi i aktoreve

Aktorët përfaqësojnë njësi të jashtme që ndërveprojnë me sistemin. Një aktor mund të jetë një human ose një sistem i jashtëm. Në shembullin SatWatch, përdoruesi i orës, satelitet GPS, WebifyWatch janë aktorë. Ata shpërndajnë informacion me SatWatch. Ndërveprimet që ata kanë me SatWatch janë të ndryshme: përdoruesi i orës e mban dhe e shikon herë pas here orën, ora monitoron sinjalet nga satelitet GPS, WebifyWatch shkarkon të dhëna të reja tek ora. Aktorët përcaktojnë klasa funksionalitetesh.

FRIEND është një shembull me kompleks. Ai është një sistem i shpërndarë informacioni për menaxhimin e aksidenteve. Ai përfshin shumë aktore, si FieldOfficer, që përfaqeson policinë dhe oficeret e zjarrit që po i përgjigjen aksidentit, Dispatcher- personi përgjegjës për t'iu përgjigjur kërkesave në numrin e urgjencës dhe për shpërndarjen e burimeve në një aksident. FRIEND ndihmon të dy aktoret duke mbajtur një plan të detyrave, histori të incidenteve dhe burimeve. Ai ka akses tek shumë baza të dhënash si psh me procedura veprimesh urgjencë. Aktoret FieldOfficer dhe Dispatcher ndërveprojnë nepermjet ndërfaqeve të ndryshme: FieldOfficer akseson sistemin FRIEND nepermjet një asistenti personal mobile, Dispatcher akseson FRIEND nepermjet një stacioni pune.



Aktoret janë abstragime rolesh dhe nuk lidhen detyrimisht drejtperdrejt me persona. I njëjti person mund të luajë rolin e FieldOfficer ose të Dispatcher në momente të ndryshme. Megjithatë, funksionalitetet që ata aksesojnë janë të ndryshme. Prandaj këto role modelohen si aktore të ndryshme.

Hapi i parë i nxjerrjes së kërkesave është identifikimi i aktoreve. Kjo shërben për të përcaktuar kufinjat e sistemit dhe për të identifikuar të gjitha perspektivat nga të cilat duhet të shohin sistemit zhvilluesit.

Në hapat e parë të identifikimit të aktoreve, shpesh është e vështirë të dallohen aktoret nga objektet. Psh një bazë të dhënash mund të jetë aktor në disa raste dhe pjesë e sistemit në raste të tjera. Kur përcaktohet kufiri i sistemit nuk është e vështirë të bëhet dallimi mes aktoreve dhe komponenteve të sistemit si objektet apo nensistemet. Aktoret janë jashtë kufirit të sistemit, ata janë të jashtëm. Nensistemet dhe objektet janë brenda kufirit të sistemit, ata janë të brendshëm. Prandaj, çdo sistem i jashtëm që përdor sistemin që do të ndërtohet është një aktor. Pyetje që behen gjatë identifikimit të aktoreve janë:

- Cilat grupe përdoruesish mbeshteten nga sistemi në punën e tyre?
- Cilat grupe përdoruesish ekzekutojnë funksionalitetet kryesore të sistemit?
- Cilat grupe përdoruesish kryejnë funksionalitete dytesore, si mirëmbajtja apo administrimi?
- Me çfarë hardware dhe software të jashtëm do të ndërveprojë sistemi?

Në shembullin e FRIEND, pyetjet më lart çojnë në një listë të gjatë aktoresh potenciale: zjarrefikesi, oficeri i policisë, hetuesi, administruesi i sistemit, etj.

Pasi identifikohen aktoret, hapi tjetër i nxjerrjes së kërkesave është përcaktimi i funksionaliteteve që do të jenë të aksesueshme për çdo aktor. Ky informacion mund të nxirret duke përdorur skenaret dhe duke formalizuar rastet e përdorimit.

2.5.2 Identifikimi i skenareve

Një skenar është një 'përshkrim narativ i asaj që njërezit mund të bëjnë duke përdorur kompjuter apo aplikim'. Një skenar është një përshkrim informal i detajuar, konkret i një karakterisitke të caktuar të sistemit nga pikepamja e një aktori të vetëm. Skenaret nuk zëvendësojnë

rastet e perdorimit. Ato perqendrohen ne instanca specifike dhe ne ngjarje konkrete. Ato jane te rendesishem sepse jane nje mjet i kuptueshem nga klienti.

<i>Emri i skenarit</i>	<u>Magazine ne zjarr</u>
<i>Instanca te aktoreve pjesemarres</i>	<u>Bob, alice: FieldOfficer</u> <u>John:Dispatcher</u>
<i>Rrjedha e ngjarjeve</i>	<ol style="list-style-type: none"> 1. Bob, nderkohe qe po patrullon ne rrugen kryesore, veren tym qe vjen nga magazine. Partneri i tij, Alice, aktivizon funksionin 'Raporto Urgjence' nga laptopi i saj ne sistemin FRIEND. 2. Alice jep adresen, nje pershkrim te vendndodhjes dhe nivelin e urgjences. Ajo konfirmon inputin dhe prët per njoftim. 3. John, dispatcher, njoftohet per urgjence nga nje tingull beep ne stacionin e tij te punes. Ai shikon informacionin e plotesuar nga Alice dhe kthen pergjigje qe e mori kete raport. Ai cakton nje njesi zjarri dhe dy njesi te tjera ne zone dhe njofton Alice per kohen kur ata mund te arrijne. 4. Alice merr njoftimin.

Skenari me lart eshte konkret, ai pershkruan nje rast specifik. Ai nuk perpiqet te pershkruaje te gjitha situatat e mundshme ne te cilat mund te raportohet nje rast zjarri. Vecanerisht, skenaret nuk mund te permbajne pershkrime te vendimeve. Per te pershkruar rezultatin e nje vendimi, duhet te perdoren te pakten dy skenare, njeri qe pershkruan rastin kur kushti eshte i vertete dhe tjetri kur kushti nuk eshte i vertete.

Gjate procesit te identifikimit te skenareve mund te behen keto pyetje:

- Cilat jane punet qe aktori pret qe te behen nga sistemi?
- Cfare informacioni akseson perdoruesi? Kush krijon te dhenat? A mund te ndryshohen apo fshihen ato? Nga kush?
- Per cfare ndryshimesh te jashtme duhet te informoje sistemin aktori? Sa shpesh? Kur?
- Per cilat ngjarje duhet aktori te informoje sistemin? Cili eshte afati qe ka?

Zhvilluesit perdorin dokumenta te domainit te aplikimit (manual perdorusisht te sistemeve te meparshem, manuale te procedurave, standarte te kompanise, intervista me klientet e perdoruesit) per t'iu pergjigjur ketyre pyetjeve.

Ne shembullin FRIEND identifikohen kater skenare qe pershkruajne tipin e detyrave qe priten nga sistemi:

- *zjarrNeMagazine*: ne magazine ka rene zjarr. Dy oficere terreni arrijne dhe kerkojne per burime.
- *Aksident*: nje aksident ndodh ne autostrade. Oficeret e policies dokumentojne aksidentin dhe menaxhojne trafikun derisa makinat e demtuara largohen nga vendngjarja.
- *maceNePeme*: nje mace ka ngecur ne nje peme. Kerkohet nje zjarrfikes per te nxjerre macen. Mqs rasti eshte me perparesi te ulet, makina vonon dhe, nderkohe macja bie nga pema dhe thyen kembet. Prandaj kerkohet dhe nje ambulance.
- *Termet*: nje termet demton seriozisht ndertesat dhe rruget, duke shkaktuar sh incidente dhe aktivizimin e nje plani veprimi per urgjenca ne rang shteteror. Njoftohet guvernatori.

Identifikimi i aktoreve dhe i skenareve ka si qellim kryesor identifikimin e domainit te aplikimit. Rezultati eshte nje kuptim me i mire i hapesires se sistemit dhe i procesit te punes qe duhet te mbeshtet nga sistemi. Skenaret formalizohen ne raste perdorimi kur ata dhe aktoret jane identifikuar dhe pershkruar.

2.5.3 Identifikimi i rasteve te perdorimit

Skenari eshte nje instance e nje rasti perdorimi, dmth nje rast perdorimi specifikon te gjitha skenaret e mundshem per nje funksionalitet te caktuar. Nje rast perdorimi iniciohet nga nje aktor dhe me pas mund te komunikojë me aktore te tjere. Nje rast perdorimi perfshin rrjedhen e plote te ngjarjeve qe nga incializimi deri ne perfundim.

<i>Emri i rastit te perdorimit</i>	RaportoUrgjence
<i>Aktoret pjesemarres</i>	Iniciohet nga FieldOffiecer Komunikon me Dispatcher
<i>Rrjedha e ngjarjeve</i>	<ol style="list-style-type: none"> 1. FieldOfficer aktivizon funksionin “Raporto Urgjence” nga terminali 2. FRIEND pergjigjet duke i shfaqur nje forme FiledOfficer. 3. FieldOfficer ploteson formen duke zgjedhur nivelin e urgjences, tupin, vendndodhjen, dhe nje pershkrim te thjeshte te situates. Ai pershkruan gjithashtu zgjidhje te mundshme per gjendjen e urgjences. Kur forma plotesohet teresisht, FieldOfficer e poston kete informacion. 4. FRIEND merr formen dhe njofton Dispatcher. 5. Dispatcher rishikon informacionin e derguar dhe krijon nje Incident ne bazen e te dhenave duke therritur rastin e perdorimit OpenIncident. Dispatcher zgjedh nje pergjigje dhe njofton qe e mori raportin. 6. FRIEND shfaq nje njoftim dhe pergjigjen e duhur ndaj FieldOfficer.
<i>Parakushtet</i>	FieldOfficer eshte i loguar tek FRIEND.
<i>Kushtet e daljes</i>	<ul style="list-style-type: none"> • FieldOfficer ka marre nje pergjigje nga Dispatcher, ose • FieldOfficer ka marre nje shpjegim qe tregon pse nuk eshte kryer veprimi I pritur normalisht.
<i>Kerkesa cilesore</i>	<ul style="list-style-type: none"> • FieldOfficer njoftohet qe kerkesa e tij eshte procesuar ose jo Brenda 30 sekondash. • Pergjigja vjen jo me vone sesa 30 sekonda qe nga momenti kur eshte derguar nga Dispatcher.

Shembull i rastit te perdorimit RaportUrgjence.

Pergjithesimi i skenareve dhe identifikimi i rasteve te perdorimit qe sistemi duhet te suportohen nga sistemi, ndihmon ne percaktimin e hapesires se sistemit. Fillimisht, identifikohen rastet e perdorimit, lidhen me aktoret perkates, emertohen dhe pershkruhen ne nivel te larte. Emri i nje rasti perdorimi duhet te jete nje grup foljor qe tregon ate c’ka perdoruesi perpiqet te arrije nepermjet tij. Psh rasti me siper nuk mund te quhet “Regjistro Urgjence” sepse emri duhet te pasqyroje perspektiven e perdoruesit dhe jo te sistemit. Ndersa “Regjistro Urgjence” tregon qe nje aktor po perpiqet te raportoje nje urgjence tek sistemi (dhe konkretisht tek aktori Dispatcher). Gjithashtu, rasti nuk mund te quhet as “perpjekje per te raportuar urgjence” sepse emri duhet te pasqyroje qellimin e rastit te perdorimit dhe jo aktivitetin aktual.

Lidhja e rasteve te perdorimit me aktoret mundeson qe zhvilluesit te identifikojne dhe qartesoje rolet e perdoruesve te ndryshem. Shpesh, duke u perpjekur per te idenfikuar se kush inicion nje rast perdorimi, zhvilluesit gjejne aktore te rinj, qe jane fillimisht ishin lene jashte.

Pershkrimi i rasteve te perdorimit perfshin pershkrimin e kater fushave. Pershkrimi i kushteve te hyrjes dhe te daljes lejon qe zhvilluesit te kuptojne cilat jane kushtet qe duhet te plotesohen qe nje rast perdorimi te therritet dhe cilat jane ndryshimet qe ai i shkakton mjedisit dhe sistemit. Gjithashtu, zhvilluesit arrijne te identifikojne nese kane anashkaluar raste perdorimi eventuale. Psh, nqs nje rast perdorimi kerkon qe te aktivizohet nje plan urgjence ne rastet e termeteve, atehere modeli i rasteve te perdorimit duhet te marre parasysh dhe nje rast per

aktivizimin e ketij plani. Pershkrimi i rrjedhes se ngjarjeve te nje rasti perdorimi nxit diskutimin mes klienteve dhe zhvilluesve per nderveprimin mes aktoreve dhe sistemit. Kjo ndikon shume ne percaktimin e kufinjve te sistemit, sepse percakton cilat veprime kryen nga perdoruesi dhe cilat nga sistemi. Pershkrimi i kushteve cilesore ndihmon nxjerrjen e kerkesave jo funksionale.

Shkrimi i rasteve te perdorimit eshte nje proces jo I thjeshte, qe permiresohet me kalimin e kohes. Disa principe qe ndihmojne ne shkrimin cilesor te rasteve te perdorimit jane:

- Rastet e perdorimit duhet te emertohen me grupe foljore. Emertimi duhet te tregojte ate qe perdoruesi perpiqet te arrije nepermjet ketij rasti perdorimi.
- Aktoret duhet te emertohen me grupe emerore.
- Kufiri i sistemit duhet te percaktohet ne menyre te qarte. Duhet te behet dallimi mes hapave qe kryen nga sistemi dhe atyre qe kryen nga perdoruesi gjate nderveprimit me njeri tjetrin.
- Hapat qe pershkruajne rrjedhen e ngjarjeve duhet te pershkruhen ne forme aktive. Ne kete menyre shprehet ne menyre eksplicite se kush ben cfare.
- Rasti i perdorimit duhet te permbaje veprimet e perdoruesist te plota (psh ne rastin e RaportoUrgjence, duhet te permbaje te tera veprimet qe kryhen qe nga moment i inicializimit te rastit te perdorimit e deri sa merr pergjigje nga sistemi).
- Rastet e vecanta (exceptions) duhet te pershkruhen ne menyre te qarte.
- Nje rast perdorimi nuk duhet te pershkruaje nderfaqen grafike te sistemit.
- Nje rast perdorimi nuk duhet te jete me i gjate sesa tre faqe.

<i>Emri I rastit te perdorimit</i>	<i>Aksident</i>	<i>Emertim jo i mire:cfare do qe te beje perdoruesi me kete rast perdorimi?</i>
<i>Aktori Iniciues</i>	Inicohet nga FieldOfficer	
<i>Rrjedha e ngjarjeve</i>	1. FieldOfficer raporton nje aksident. 2. Dergohet nje ambulance. 3. Dispatcher njoftohet kur ambulance arrin ne vend.	<i>Cili veprim beri qe FieldOfficer te marre nje njoftim?</i> <i>Forme pasive, kush e shperndan ambulancen?</i> <i>Pershkrim jo I plote i rrjedhes se veprimeve: cfare ben FieldOfficer pasi ambulance eshte shperndare?</i>

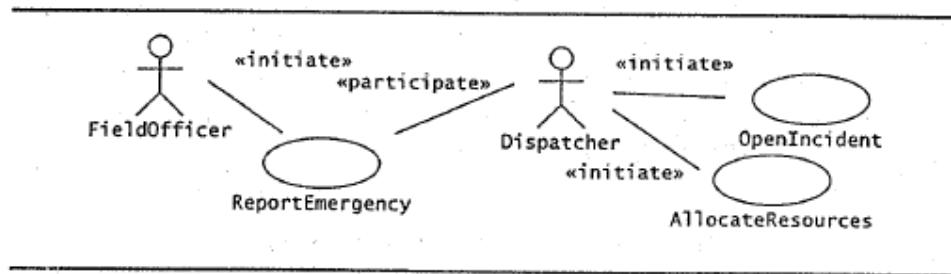
2.5.4 Identifikimi i lidhjeve ndermjet aktoreve dhe rasteve te perdorimit

Prania lidhjeve ndemjet aktoreve dhe rasteve te perdorimit mundeson qe zhvilluesit dhe perdoruesit te ulin kompleksitetin e modelit dhe te rrisin kuptueshmerine e tyre. Lidhjet komunikuese perdoren per te treguar lidhjet mes aktoreve dhe rasteve te perdorimit dhe e ndajne sistemin ne shtresa funksionaliteti. Lidhjet *zgjero (extend)* perdoren per te ndare rastet e vecanta te rrjedhes normale te ngjarjeve. Lidhjet *perfshi (include)* perdoren per te reduktuar teprine e rasteve te perdorimit.

Lidhjet komunikuese ndermjet aktoreve dhe rasteve te perdorimit

Lidhjet komunikuese paraqesin rrjedhen e informacionit ne nje rast perdorimi. Mund te identifikohet dhe vecohet aktori qe inicion rastin e perdorimit nga ata qe vetem komunikojne me te. Duke specifikuar ata aktore qe komunikojne me rastin e perdorimit, ne menyre jo direkte identifikohen dhe ata qe nuk komunikojne me te. Dokumentimi i lidhjeve te inicimit dhe te

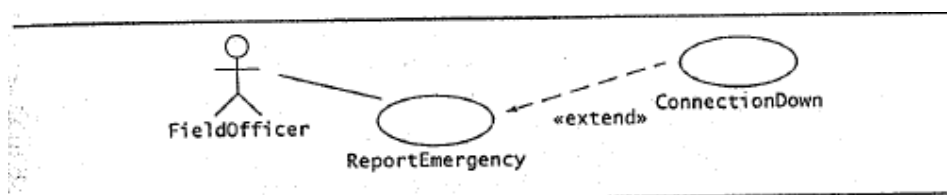
komunikimit ndermjet aktoreve dhe rasteve te perdorimit, specifikon edhe kontrollin e aksesit ne nivel sistemi.



Shembull i lidhjeve te komunikimit mes aktoreve dhe rasteve te perdorimit te FRIEND.

Lidhjet zgjero (extend) ndermjet rasteve te perdorimit

Nje rast perdorimi zgjeron nje tjeter nqs rasti i pare perfshin sjelljen shtese ne raste apo kushte te caktuara. Ne rastin e sistemit FRIEND, mund te ndodhe qe linja e komunikimit mes FieldOfficer dhe Dispatcher te shkeputet nderkohe qe FieldOfficer po ploteson formen e raportimit. FieldOfficer duhet te njoftohet nga stacioni i tij i punes qe forma me informacionin qe ai plotesoi nuk u dergua dhe duhet ta njoftoje gjithashtu se cfare ai duhet te beje.



Shembull i lidhjeve zgjero (diagrame UML). ConnectionDown zgjeron rastin ReportEmergency. ReportEmergency perqendrohet vetem ne raportimin e urgjences.

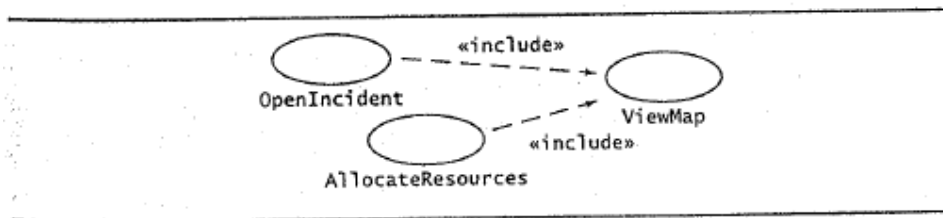
Ndarja e veprimeve ne rastet e vecanta nga ato ne rastet normale ka dy avantazhe:

- rasti i zakonshem i perdorimit behet me i thjeshte dhe me i shkurter
- rasti i zakonshem ndahet nga rasti i vecante, gje qe ben qe zhvilluesit te trajtojne cdo tip funksionaliteti ne menyra te ndryshme.

Te dyja rastet e perdorimit, ai normal dhe ai i vecante, jane raste perdorimi me vete. Secili duhet te kete parakushte dhe kushte daljeje, dhe duhet te jene te kuptueshme nga perdoruesi.

Lidhjet 'perfshi' ndermjet rasteve te perdorimit

Lidhjet 'perfshi' perdoren per te faktorizuar teprite qe ekzistojne ndermjet lidhjeve te perdorimit. Psh. Dispatcher ka nevojte te shikojte harten e qytetit kur hap nje incident ne sistem dhe kur percakton shperndarjen e burimeve (keshtu mund te percaktoje psh se cili burim eshte me afer vendit te incidentit). Ne kete rast, ViewMap eshte nje rast perdorimi qe pershkruan rrjedhen e ngjarjeve qe duhen kur shikohet harta e qytetit. Ajo kerkohet nga te dyja rastet, nga OpenIncident dhe AllocateResources.



Shembull i lidhjes pefshi

Lidhjet *zgjero* perdoren ne rastet kur ka sjellje ne kushte te vecanta, qe nuk ndodhin shpesh dhe qe nuk jane te zakonshme. Psh. mosdisponimi i nje burimi te caktuar per nje interval te shkurter.

Lidhjet *perfshi* per te treguar sjellje qe jane te perbashketa per dy ose me shume raste perdorimi.

2.5.5 Identifikimi i kerkesave jo funksionale

Kerkesat jo funksionale shprehin aspekte te sistemit qe nuk lidhen drejperdrejt me funksionet e tij. Ato perfshijne qe nga pamja e nderfaqes grafike deri tek koha e pergjigjes apo ceshtjet e sigurise. Kerkesat jo funksionale percaktohen ne te njejten kohe me kerkesat funksionale sepse ato kane nje ndikim te madh te l gjithe procesi i zhvillimit dhe tek kostoja.

Per te nxjerre kerkesat jo funksionale, klientet dhe zhvilluesit duhet te bashkepunojne per te identifikuar atributet e sistemit qe jane te veshtira per t'u kuptuar dhe qe kane ndikim te rendesishem ne punen e perdoruesit.

Bashkesia e kerkesave jo funksionale permban shpesh kerkesa kontradiktore. Psh ne rastin e SatWatch, kerkesat jo funksionale kerkojne nje ore me mekanizem te sakte ne menyre qe ora te mos rregullohet nga perdoruesi dhe me kosto te ulet, ne menyre qe te zevendesohet lehtesisht nqs thyet. Keto dy kerkesa jane kontradiktore, sepse kostoja e ores rritet kur saktesia e saj eshte e larte. Per te menaxhuar keto raste, klienti dhe zhvilluesit i caktojne nje prioritet kerkesave jo funksionale. Kjo siguron qe kerkesat te trajtohen me konsistence.

Kerkesa	Pyetje te mundshme
Perdorueshmeria	<ul style="list-style-type: none"> Cili eshte niveli i ekspertizes se perdoruesit? Me cilat standarte te nderfaqes eshte familiar perdoruesi? Cfare dokumentimi duhet t'i jepet perdoruesit?
Besueshmeria	<ul style="list-style-type: none"> Sa i besueshem, i disponueshem duhet te jete sistemi? A eshte i pranueshem eshte ristartimi ne raste problemesh? Sa te dhena lejohet qe te humbase sistemit? Si duhet t'i trajtoje sistemi rastet e vecanta? A ka kerkesa sigurie per sistemin?
Performanca	<ul style="list-style-type: none"> Sa shpejt duhet te pergjigjet sistemi? A ka funksione kritike nga pikepamja kohore? Sa perdorues qe e perdorin njekohesisht duhet te suportoje sistemi? Sa eshte vonesa me e madhe e lejueshme per perdoruesit?

Pyetje te mundshme per nxjerrjen e kerkesave

2.6 Dokumentimi i kerkesave

Rezultati i nxjerrjes se kerkesave dhe i analizes se tyre dokumentohet ne Dokumentin e Analizes se Kerkesave (DAR). Ky document pershkruan sistemin nepermjet kerkesave funksionale dhe jo funksionale dhe sherben si nje baze kontraktuale ndermjet klientit dhe zhvilluesve. Audienca e RAD perfshin klientet, perdoruesit, menaxhimin e projektit, analistet e sistemit, modeluesit e sistemit, Pjesa e pare e dokumentit perfshin rastet e perdorimit dhe kerkesat jo funksionale dhe shkruhet gjate nxjerrjes se kerkesave. Formalizimi i specifikimeve ne objekte modelesh shkruhet gjate analizes.

Dokumenti i Analizes se Kerkesave

1. Hyrje
 - 1.1 Qellimi i sistemit
 - 1.2 Hapesira (scope)e sistemit
 - 1.3 Objektivat dhe kriteret e suksesit te projektit
 - 1.4 Perkufizime, shkurtime dhe akronime
 - 1.5 Referenca
 - 1.6 Permbledhje
2. Sistemi aktual
3. Sistemi i propozuar
 - 3.1 Permbledhje
 - 3.2 Kerkesa Funksionale
 - 3.3 Kerkesa Jo Funksionale
 - 3.3.1 Perdorueshmeria
 - 3.3.2 Besueshmeria
 - 3.3.3 Performanca
 - 3.3.4 Mbeshtetja
 - 3.3.5 Nderfaqe
 - 3.4 Modeli i sistemit
 - 3.4.1 Skenaret
 - 3.4.2 Rastet e perdorimit
 - 3.4.3 Modeli i objekteve
 - 3.4.4 Nderfaqja me perdoruesin – linja e preferuar e navigimit ne ekran dhe prototype te nderfaqes
4. Fjalor

Paragrafi *Hyrje* jep nje permbledhje te shkurter te funksioneve te sistemit, te arsyeve per zhvillimin e sistemit, hapesiren e tij dhe te referencave per kontekstin e zhvillimit (psh. referenca tek deklarimi i problemit i pershkruar nga klienti, tek sisteme ekzistuese, studime te aftesive praktike). Ketu perfshihen edhe objektivat dhe kushtet e suksesit te projektit.

Paragrafi *Sistemi aktual* pershkruan gjendjen aktuale. Nqs sistemi I ri do te zevendesojte nje ekzistues, atehere ky sistem pershkruan funksionalitet dhe problemet e sistemit aktual. Perndryshe, ky paragraph pershkruan sesi kryen per momentin detyrat qe do te mbuloje sistemi i ri.

Paragrafi *Sistemi i propozuar* dokumenton nxjerrjen e kerkesave dhe modelin e analizes se sistemit te ri. Ai ka kater nenparagrafe:

- Permbledhje: permbledhje e kerkesave funksionale te sistemit
- Kerkesat funksionale: pershkruan funksionalitetet ne nivel te larte

- Kerkusat jofunksioanale: kerkesa te perdoruesit qe nuk lidhen drejperdrejt me funksionalitetet.
- Modeli i sistemit: pershkruan skenare, raste perdorimi, modelin e objekteve. Ky seksion permban nje specifikim te plote te kerkesave funksionale duke perfshire modele (mock-up) te nderfaqes grafike, linjen e navigimit neper format e nderfaqes grafike.

DAR duhet te shkruhet pasi modeli i rasteve te perdorimit eshte stabilizuar dhe eshte i qendrueshem, pra atehere kur ndryshimet qe i behen kerkesave jane minimale. Ne seksionin e historise se rishikimeve qe i behen dokumentit, dokumentohen autori i ndryshimeve, datat e ndryshimeve dhe pershkrime te shkurtra te ndryshimeve te bera.

3. Analiza

Analiza perqendrohet ne krijimin e nje modeli te sistemit, qe quhet modeli i analizes, i cili eshte i sakte, i plote, konsistent dhe pa dykuptimesi. Ndryshimi i analizes nga nxjerrja e kerkesave eshte se gjate analizes zhvilluesit perqendrohen ne strukturimin dhe formalizimin e kerkesave te nxjerra nga perdoruesit. Ky formalizim ndihmon ne kuptimin me ne thellesi te kerkesave dhe ne zbulimin e gabimeve te ndryshme. Dokumenti i specifikimit te kerkesave duhet te ndryshohet vazhdimisht nga zhvilluesit per te pasqyruar ato c'ka jane kuptuar me mire apo jane korrigjuar ne lidhje me kerkesat. Kjo sepse modeli i analizes nuk eshte shume i kuptueshem per perdoruesit, nderkohe qe dokumenti i specifikimit te kerkesave eshte me i lexueshem per ta.

Modeli i analizes ka tre komponente kryesore:

1. modeli funksional: perbehet nga rastet e perdorimit dhe skenaret. Ekzaminimi i secilit rast perdorimi con ne identifikimin e objekteve te mundshme
2. modeli i objekteve te analizes: perbehet nga diagramat e klasave. Perqendrohet ne konceptet qe manipulohen nga sistemi, vecorite dhe lidhjet e tyre. Ai sherben si nje fjalor vizual i koncepteve kryesore qe jane te dukshme per perdoruesin.
3. modeli dinamik: perfaqesohet nga diagramat e gjendjes dhe nga diagramat e sekuences. Ai perqendrohet ne sjelljen e sistemit. Diagramat e sekuences paraqesin nderveprimet mes objekteve te ndryshem te nje rasti perdorimi. Diagramat e gjendjes paraqesin sjelljen e nje objekti te vetem (ose te nje grupi shume te vogel dhe me lidhje te ngushta objektesh). Modeli dinamik sherben per tu caktuar pergjegjesite e duhura seciles klase dhe per te identifikuar (gjate procesit te ndertimit te tij) klase te reja, lidhje dhe attribute qe duhen shtuar ne modelin e objekteve te analizes.