

**Priprava izhodišč za pripravo zahtev pri implementaciji in razvoju IKT rešitev –
Operativno/delovno gradivo:**

**Osnovna izhodišča za pripravo zahtev in izvedbo IKT rešitev in sistemov na
segmentu OT**

Pripravil: Kristjan Cah
Področje: Upravljanje, Oddelek za cestne naprave
Sistem: Sistemi za upravljanje s prometom
Zadnja sprememba: 3.1.2022
Verzija dokumenta: 1.5

1. UVOD - USMERITVE IN ZAHTEVE

V dokumentu obravnavamo ključne usmeritve, predstavljene v obliki splošnih, funkcionalnih in delno tehničnih zahtev in so pogoj za integracijo in centralizacijo IS in rešitev pri naročniku. Nekatere rešitve in usmeritve temeljijo na vsebinah in tehnologijah ki so že prisotne, splošni namen pa je usmeritev vsem deležnikom, da na enak način razumejo in delujejo v smeri skupnega cilja, to je optimalnega nivoja centralizacije in integracije sistemov naročnika. Usmeritve morajo **upoštevati načrtovalec in pripravljavec dokumentacije za izvedbo, izvajalec in interni strokovnjaki naročnika**. Pri izdelavi dokumentacije, načrtov in rešitev je potrebno upoštevati tudi zunanje okolje, najpogosteje v obliki zakonodaje in uredb, poslovni nivo zahtev skladno z vizijo in strategijo družbe in tehnološkim oziroma operativnim nivojem, skladnost z dobrimi praksami in standardi na obravnavanem področju. Za celovito obravnavo je potrebno z zahtevami pokriti nivoje oskrbne infrastrukture, sistemov hlajenja, fizične in logične varnosti, vse nivoje OSI komunikacijskega modela. V dotičnem poglavju pa smo se opredelili le na sistemsko programsko opremo in aplikativne programske rešitve. Izpostavljene usmeritve v poglavju predstavljajo povzetek oziroma izvleček za snovalce sistemov in rešitev (projektante) in tudi izvajalce ob implementaciji. Usmeritve in zahteve so sestavni del celovitega dokumenta smernic za integracijo in centralizacijo sistemov v nadzornih centrih za upravljanje s cestnim prometom.

2. SPLOŠNE ZAHTEVE

V poglavju so našteje nekatere splošne zahteve, katere predstavljajo osnovo zahtevanih elementov obravnavanih v dokumentaciji za izvedbo in kateri predstavljajo vodilo za projektanta in ostale na segmentu IKT programskih rešitev in sistemov in so temelj za projektne naloge, kjer se projektira tudi IKT rešitve. Rešitve morajo zagotavljati oziroma predvideti/obravnavati vsaj:

- **Metodologijo:** Projektant skupaj z naročnikom, glede na vsebino zahtevanih funkcionalnih in tehničnih zahtev, opredeli ustrezno metodologijo življenjskega cikla sistema in posledično razvoja programske rešitve. Metodologija mora poudariti v fazi izvedbe principe dela in potrebnih interakcij, posamezne faze razvoja, ključne mejnike, tipe dokumentacije in obseg le te, itn. Opredeljeno metodologijo mora argumentirati s pričakovanimi učinki in prednostmi napram drugim pristopom. Metodologija mora predvideti intenzivno sodelovanje in prisotnost naročnika v vseh fazah tako projektiranja kot tudi izvedbe (zaželeno so agilne metodologije).
- **Osnovni principi arhitekture:** Vsi koncepti predvidene arhitekture rešitve morajo biti usklajeni s poslovnimi arhitekti naročnika, kateri na osnovi ocene tveganja rešitve in celotnega IKT okolja, potrjujejo vsebino in morebitna odstopanja od ciljne arhitekture.
 - Arhitekture rešitev morajo ciljno zasledovati oziroma upoštevati relevantne lastnosti nekaterih drugih arhitektur kot so realno časne systemske arhitekture, dogodkovno vodene arhitekture, varnostne arhitekture kritičnih sistemov, storitveno orientirane arhitekture, računalništvo na robu, računalništvo v megli, decentralizirane in podatkovno vodene arhitekture. Pri najbolj kritičnih sistemih pa uporaba arhitekture digitalnih dvojčkov.
 - V osnovi mora arhitektura temeljiti na tri ali več nivojski arhitekturi, kjer ločujemo vsaj podatke od poslovne logike in uporabniške vmesnike od poslovne logike.
 - Upoštevati je potrebno arhitekturno uporabo skupnih gradnikov za sorodne funkcionalnosti, kot so osnovna systemska platforma z virtualizacijskim okoljem in požarnimi pregradami, sistemi upravljanja identitet in dostopov, integracijske platforme, nadzora oziroma monitoringa naprav in grafičnih vmesnikov.
 - Rešitev mora predvidevati možnost integracije, management platforme, na primer API management, za povezljivost notranjih in zunanjih sistemov in aplikacij, kjer je poudarek na enotnih integracijskih protokolih, vmesnikih, varnostnih pravilih, konceptih obvladovanja in spremljanja le teh.

3. ARHITEKTURNA IZHODIŠČA

V nadaljevanju je podanih še nekaj ključnih zahtev za posamezne segmente informacijskih sistemov in rešitev:

- **Neodvisnost:** Izdelane in predane rešitve morajo biti izvedene in predane na način, da ne pogojujejo odvisnosti naročnika od trenutnega dobavitelja (ang. »vendor lock-in«). V primeru rešitev, katere so licenčne, delno licenčne
- **Statični model sistema,** kateri predstavlja osrednjo komponento, ki omogoča opis vsake naprave, senzorja ali opreme, ki je del obravnavanega sistema (npr. NKS, SNVP, VNP, VDP,...). Opis mora opredeliti glavne lastnosti naprave, tip naprave, ključne podatke o napravi, podatkovno strukturo s katero naprava operira, protokole, standarde, geografsko lokacijo naprave in druge opisne podatke (npr. vsaj skrbnika, leto namestitve/dobave, vzdrževalni posegi, itn). Vsaka nova naprava mora biti določenega tipa, kateri že v naprej določi njeno delovanje in lastnosti. Na ta način se opredeli naprave in njihov način delovanja teh hitrejši vključevanje v produkcijsko delovanje. Lastnosti določenega tipa naprave se tako upravljajo centralizirano na enovitem mestu, od koder se nato distribuirajo do komponent sistema, ki take konfiguracije potrebujejo za svoje delovanje.
- **Standardi in dobre prakse:** Informacijski sistem mora uporabljati uveljavljene, preizkušene in sodobne standarde na področju varnosti, kakovosti programske opreme izmenjave podatkov, integracij, API vmesnikov, komunikacij ter specifičnih EU standardov na področju sistemov za upravljanja prometa (Primer: **OPC UA** za izmenjavo podatkov in kontrol do obcestnih naprav, DATEX II za izmenjavo podatkov z drugimi sistemi ITS), ter drugih splošnih standardov za obvladovanje kakovosti IS in posledično upravljanje življenjskega cikla programskih rešitev, kot so na primer **ISO/IEC/IEEE 29148-2011** – Sistemsko in programsko inženirstvo – Življenjski cikel – Inženirstvo zahtev; **ISO/IEC/IEEE 12207:2017** – Sistemsko in programsko inženirstvo – Proces življenjskega cikla programske opreme; **ISO/IEC/IEEE 29119-2013 01-05** Sistemsko in programsko inženirstvo – Testiranje programske opreme; **EN IEC/IEEE 82079-1-2019** Priprava informacij za uporabo – Navodila, itn. Upoštevati je potrebno uveljavljene primere dobrih praks (npr. ITIL).
- **Vrsta rešitve:** Pri snovanju rešitve, bodisi da bo ta licenčna ali razvita namensko, je potrebno vedno preveriti možnost uporabe preizkušenih in uveljavljenih odprtokodnih rešitev ali komponent, pri čemer imajo prednost tiste, ki jih podpirajo oziroma zastopajo tudi lokalni dobavitelji, lokalni integratorji in podjetja za razvoj programske opreme. Ključni kriteriji pri izbire odprtokodnih rešitev: preizkušene, upoštevanje varnostnih standardov, standardni protokoli in lokalna podpora
- **Ponovna uporaba:** Moduli, komponente, konfiguracije in vmesniki aplikativnih programskih rešitev morajo biti grajeni modularno ter dokumentirani na način, ki omogoča enostavno ponovno uporabo. Ponovna uporaba gradnikov (API) ima prednost pred nakupom ali razvojem novih komponent, istih ali podobnih funkcionalnosti. Rešitve morajo omogočati spreminjanje pogostejših parametrov delovanja brez posega v programsko kodo (parametrizacija). Pravice za spreminjanje parametrov ima praviloma administrator.
- **Skalabilnost:** Aplikativne programske rešitve in sistemi morajo biti zasnovani na način, ki omogoča enostavno prilagajanje sistema v smislu novih uporabnikov, integracije novih delovnih postaj, dodajanje strojnih in sistemskih zmogljivosti, dodajanja nove opreme in naprav, kakor tudi dodajanja novih podatkovnih tipov. Enako velja za zmanjševanje.
- **Interoperabilnost:** Aplikacijske rešitve in celotna arhitektura morajo zagotavljati semantično in tehnično interoperabilnost na podlagi odprtih, širše sprejetih in neodvisnih standardov.
- **Centralno:** Pri načrtovanju rešitev in gradnikov arhitekture je potrebno upoštevati, da se funkcije nadzora in vodenja prometa za določeno traso ali predor izvajajo centralno, pri čemer se podatki distribuirajo (replicirajo) na več fizičnih lokacij nadzornih centrov. Kdo, kaj upravlja in s katerimi podatki ter kje, pa opredeljujejo pravice uporabnika/nadzornika. Sistemi morajo arhitekturno omogočati komponento/proces, katere skrbi za replikacijo na druge lokacije.
- **Zahtevana okolja:** Aplikativne rešitve morajo upoštevajoč njihovo kritičnost pri upravljanju pomena predvideti vsaj tri ločena okolja, ki jih mora zagotavljati tako naročnik, kot zunanji izvajalci, ki sodelujejo pri razvoju. To so Razvojna okolja (ta so lahko nameščena izključno pri izvajalcu), testno okolje kot osnova za izvedbo in potrditev

testov pred produkcijo, za najbolj kritične dele arhitekture se zahteva tudi pred produkcijsko »stage« okolje (pre-deployment), ki je vzpostavljeno in upravljano pri naročniku. Po potrebi in glede na obseg se izdelajo zahteve tudi za učno okolje. Potrebno je jasno opredeliti proces, ki določa prehajanje rešitev med okolji, odgovornosti vpletenih ter določena osnovna pravila, ki veljajo v vsakem izmed okolij.

- **Nameščanje aplikacij:** Pri nameščanju ali posodobitvah uporabniških aplikativnih programskih rešitev in komponent, je potrebno predpisati mehanizme, kateri morajo predvideti možnost nameščanja rešitev centralno, torej iz centralne lokacije, iz nadzorovanega in varnega vira, ki je prestal zahtevana testiranja. Nameščanje mora biti ustrezno dokumentirano, tako da omogoča naročniku tako namestitve, kot tudi konfiguracijo rešitve.
- **Nadzor sistemov in rešitev:** Vse aplikacije, komponente in programski moduli morajo biti zasnovani na način, ki omogoča enostavno vključitev v sistem za centralno spremljanje delovanja in diagnostiko sistemov naročnika (monitoring in diagnostika). Sistemi in rešitve morejo predvideti, da nadzorni sistem poslučuje s podatki o stanju in statusu strojne opreme, delovanju sistemskih in programskih sistemov, predvsem stopnje napak, odzivni časi, stopnje zahtev, skupno razpoložljivostjo, itn. Za navedene namene je potrebno v sistemih opredeliti ustrezne metrike in kazalnike.
- **Redundanca:** Kritični deli arhitekture sistema, kot so kritične aplikacije, skupni aplikacijski gradniki, integracije, podatki in tudi fizične lokacije za vodenje in nadzor sistema morajo biti zasnovani na način, ki omogoča redundantnost. Kritične zmogljivosti (funkcije) naj imajo možnost delovanja/preklopa v redundantnem načinu (npr. prevzem nadzora in vodenja prometa iz rezervne lokacije). Zasledovati je potrebno minimalni čas preklopa in prehod brez oz. čim krajšim izpadom.
- **Visoka razpoložljivost:** Tako na sistemskem kot aplikativnem nivoju je potrebno predvideti in predpisati mehanizme visoke razpoložljivosti (npr. vsaj sisteme za virtualizacijo, izvedba z uporabo grozdov/gruč in geografsko ločenih grozdov opreme in storitev (ang. Cluster), varnostne kopije in postopki obnove, periodične vaje obnove).
- **Zmogljivost:** Tehnične specifikacije, katere opredeljujejo zmogljivosti sistema, tako strojnih kot programskih segmentov, morajo zajemati izračune oziroma elemente, kateri empirično opredeljujejo potrebo po posamezni opredeljeni zmogljivosti. Pri načrtovanju in posledično razvoju rešitev, storitev in aplikacij znotraj sistema za nadzor in vodenje prometa je potrebno stremeti k racionalni uporabi računalniških virov (pomnilniške kapacitete, procesorski viri, komunikacije). Zahteva se optimalna utilizacija strežniških virov, kar v praksi pomeni sobivanje več aplikacij, komponent ali podatkovnih baz sistema na istem strežniku (fizičnem ali virtualnem) do meje, ki pomeni še sprejemljivo zanesljivost, varnost in odzivnost sistema, upoštevajoč varnostni faktor. Okolje virtualizacije obvladuje naročnik.

4. VARNOSTNI VIDIKI

Informacijski sistemi morajo upoštevati vse vidike standardov in dobrih praks na segmentu informacijske varnosti. Upoštevati je potrebno certifikacijo naročnika s standardom 27001.

- **Upravljanje identitet:** Aplikativne programske rešitve morajo biti zasnovane tako, da omogočajo uporabo mehanizmov centraliziranega upravljanja identitet. Kjer ni objektivnih razlogov za drugačno upravljanje identitet, morajo vsi uporabniki in aplikacije uporabljati enoten sistem za upravljanje identitet in avtentikacij ter avtorizacij naročnika, ter s tem povezane in sprejete standarde, protokole in nosilce, npr. registrirne kartice naročnika. Vsa odstopanja morajo biti predhodno usklajena in potrjena s strani naročnika.
- **Sistemska varnost:** Novo izdelani in nadgrajeni sistemi in rešitve morajo zagotavljati možnost uporabe protokolov v okviru in povezavi z Microsoft domeno. Sistemi morajo omogočati vsaj LDAP protokole za komunikacijo z aktivnim imenikom, SSL protokole šifriranja, varnostne in skupinske politike, NTLM in Kerberos preverjanje pristnosti, Unix sistemi SMB/CIFS protokole za deljenje virov in LDAP integracijo (PAM modul).
- **Avtentikacija in avtorizacija:** Pri avtentikaciji je potrebno glede na kritičnost rešitve, kot tudi glede na pravice uporabnika oziroma administratorja, določiti koliko faktorjev/nivojev avtentikacije se bo uporabilo. Pri kritičnih sistemih se določi vsaj dvo-faktorsko ali celo več faktorsko avtentikacijo, o lastnostih posameznih faktorjev pa se

projektant ali izvajalec uskladita z naročnikom. Predvidena rešitev oziroma sistem morata zagotavljati možnost priklopa na sistem **enotne/enkratne avtentikacije in avtorizacije** (angl. Single sign on) in tako omogočati sistem kateri namenska rešitev centralno upravlja in nadzoruje dostope in pravice posameznega uporabnika.

- **Kontrole dostopa:** V okviru rešitev mora projektant predvideti tako logično kot tudi fizično varnost. Iz navedenega je potrebno predvideti, da se v novo izdelanih ali nadgrajenih sistemih, zagotovi fizična kontrola pristopa do ključne opreme, katera mora uporabljati navedene mehanizme avtentikacije in avtorizacije iz prejšnjih alinej.
- **Požarni zidovi:** Pri določanju vključevanja sistemov in rešitev v okolje naročnika je potrebno upoštevati, da razen izjemoma in z dovoljenjem naročnika, ni dovoljeno umeščanje dodatnih požarnih pregrad, ampak sisteme in rešitve vezati na obstoječo infrastrukturo naročnika, skladno z usmeritvami, katere se dogovori tekom priprave izvedbene dokumentacije.
- **Revizijska skladnost:** Za kritične aplikacijske rešitve in komponente sistema za nadzor in vodenje prometa je potrebno predpisati zagotavljanje revizijskih sledi, ki ustrezajo standardom in dobrim praksam na tem področju in lahko prestanejo revizijski pregled brez ugotovljenih nepravilnosti s strani revizorja informacijskega sistema. Zahteve stopnjo za revizijske sledi opredeli projektant in jih skupaj z naročnikom pregleda in potrdi, za vsako posamezno aplikacijo, komponento ali podsistem ločeno (npr. za najbolj kritične rešitve se predvidi poglobljena revizijska sled (ang. audit trail), ki se določi na nivoju podatkov in za vse operacije generiranja, branja, spreminjanja in brisanja (ang. CRUD), za manj kritične pa predvsem vsaj splošne informacije, kdo, kdaj in kaj je uporabljal). Skladno z zakonodajo in upravljanjem osebnih podatkov je potrebno predvideti da imajo dostope za pregled revizijskih sledi samo pooblaščen osebe, dostopi do revizijskih sledi pa se morajo ravno tako beležiti.
- **Dnevniške datoteke:** Predvideti je potrebno da vsaka programska rešitev ali sistem zagotavlja ustrezno zapisovanje dostopov, posegov in kljunih aktivnosti, skladno z dobrimi praksami in zakonskimi zahtevami (slednje velja predvsem za sisteme videonadzora). Navedena točka je glede na to da gre za robni segment med omrežnimi sistemi in sistemsko platformo, opredeljena tudi v smernicah za omrežne sisteme.
- Upravljanje varnostnih dogodkov:

5. PROCESI IN UPORABNIŠKI DEL

Sistemi in rešitve morajo poleg arhitekturnih zahtev zadostiti tudi nekaterim izhodiščem vezanim na procesne in uporabniške vidike:

- **Praviloma spletno:** Z namenom lažje integracije, dostopnosti in fleksibilnosti je potrebno tam, kjer ni objektivnih razlogov za drugačno arhitekturo, uporabniške aplikacije in uporabniške vmesnike zasnovati v smeri implementacije z uporabo spletnih tehnologij. Rešitve morajo biti dostopne preko spletnih brskalnikov širokega nabora, pri tem je potrebno predpisati uporabo najnovejših različic in posodobitve brskalnikov, čemur ustrezno mora biti zasnovana tudi programska rešitev.
- **Enostavnost:** Aplikacije morajo biti enostavne za uporabo, zadoščati uporabniškim zahtevam, biti intuitivne ter zagotavljati dobro uporabniško izkušnjo. Uporabniški vmesniki morajo biti poenoteni in morajo upoštevati vsaj naslednja načela: preprostost uporabe (hitro učenje), preglednost gradnikov, konsistentna uporaba gradnikov (isti element, isti pomen, na istem mestu), prepoznavnost uporabljenih gradnikov (npr. gumb za akcijo ima vedno enak izgled), vizualna hierarhija (uporabnik vedno ve kje se nahaja), učinkovitost uporabe (uporabnih do ključnih akcij trenutnega pogleda pride v kliku ali dveh), odzivnost (hiter odziv na uporabnikove akcije). Testiranje uporabniške izkušnje in vključevanje končnih uporabnikov v testiranje mora biti del uvajanja novih rešitev. Test uporabniške izkušnje s strani naročnika potrjuje ustreznost uporabniškega vmesnika. V primeru neskladnosti uporabniškega vmesnika lahko naročnik zahteva presojo uporabniške izkušnje s strani neodvisnega svetovalca ali presojo po standardu ISO 9241-210.
- **Varnost aplikacijskih rešitev:** Varnost podatkov in transakcij je ključnega pomena za zanesljivo in varno delovanje sistema. Varnostno načrtovanje, preverjanje in testiranje mora biti del vsake nadgradnje sistema.

Projektant mora predvideti, da izvajalci upoštevajo varnostne standarde in priporočila (npr. ISO/IEC 27001, EU ENISA priporočila za ICS SCADA za kritično infrastrukturo). Opredeliti je potrebno, katera poročila o varnostnem testiranju morajo izvajalci predati ob predaji rešitev. Varovanje pretoka podatkov in transakcij mora predvidevati varovanja tudi na logičnem sistemskem nivoju, predvsem z uporabo požarnih pregrad, sistemov za detekcijo/preprečevanje nepooblaščenih dostopov, sistemov zaščite pred zlonamerno kodo in sistemov za distribucijo varnostnih popravkov.

- **Povezovanje in aplikacijski protokoli:** Pri integraciji z zunanjimi sistemi je potrebno slediti ustrezni metodologiji razvoja in testiranja programske opreme (kot na primer kontinuiran integracijski model). Predvideti je potrebno ustrezne protokole validacije in šifriranja, kateri so najbolj običajno v tovrstnih rešitvah uporabljeni. V primerih prevzemanja podatkov iz internih in predvsem zunanjih sistemov pa je potrebno predpisati, da aplikacijski programski vmesniki sledijo dogovorjenim predpisom, zagotavljajo ustrezen nivo varnosti, zanesljivosti in njihovi upravitelji ne spreminjajo dogovorjenih lastnosti brez predhodnega dogovora.
- **Dokumentiranost rešitev:** Vse rešitve, moduli, elementi, vmesniki, aplikacije, storitve, podatkovne baze, komunikacije in naprave, ki jih dobavljajo ali implementirajo zunanji izvajalci, morajo biti dokumentirane na standardni način, predpisan s strani naročnika in usklajen s projektantom. Kot primer, vmesniki (API) morajo biti dokumentirani na enoten način (specifikacija OpenAPI, zadnja verzija 3.0). Zahtevati je potrebno, da se dokumentacija redno osvežuje in uvrščena v odlagališče (ang. repository) sistemske dokumentacije. Podrobnosti za zahtevane dokumentacije se določa skladno z metodologijo in predvidenimi standardi in dobrimi praksami, v fazi priprave dokumentacije za izvedbo skupaj z naročnikom.

6. IZHODIŠČA ZA TESTIRANJE

Testiranje je osnova za zagotavljanje kakovosti programske opreme in je ključnega pomena za stabilno in zanesljivo delovanje celotnega sistema. Testiranja morajo biti izvedena skladno z standardi (predvsem ISO/IEC 29119) in dobrimi praksami.

- **Planiranje in izvedba testiranja:** Predpisati je potrebno postopke planiranja testiranja in izvedbe testiranja na različnih nivojih (performančni testi, varnostni testi, uporabniški testi) pri vseh spremembah, nadgradnjah ali razvoju novih aplikativnih rešitev. Upoštevati je potrebno dobre prakse na področju razvoja in testiranja programske opreme kot na primer pristop DevOps, kateri v inkrementalnih ciklih zagotavljamo povratne informacije, minimalne čase prehoda med fazami razvoja, kontinuirano testiranje in odprava napak v vsakem ciklu, hitra in pogosta dostava različic
- **Samodejno testiranje:** Potrebno razmišljati tudi v smeri samodejnega testiranja v vseh segmentih, kjer je to mogoče in predvideti tudi primerna orodja, glede na tip opreme, kritičnost in zahtevane zanesljivosti, celovitosti in varnosti delovanja.
- **Dokumentacija testiranja:** Vsako testiranje mora spremljati obvezna dokumentacija. Predvideti je potrebno vso potrebno testno dokumentacijo, katero mora biti predložena ob izvedbi, vsaj pa:
 - Načrte testiranja, s pripadajočo časovnico in aktivnostmi testiranja, definicijo okolij in odgovornih oseb, dokumentov in orodij
 - Testne scenarije in testne primere, s kratkimi opisi scenarijev ali primerov z ustreznimi enoličnimi oznakami z relacijami na zahteve, pogoji in zahteve za testiranje, funkcionalnosti, itn.
 - Poročilo testiranja s splošnimi podatki o testiranju in statistiko ugotovljenih neskladij, ter rezultati testiranja
- **Testna in simulacijska okolja:** Opredeliti je potrebno tudi vsa potrebna testna in simulacijska okolja, pri katerih je zahteva da v največji meri odražajo arhitekturo produkcijskega okolja. Predvideti je potrebno:
 - Kje se izvajajo testiranja in kje postavljajo katera testna okolja (večinoma razen razvojnih aktivnosti je zahtevano, da so okolja pri naročniku.

- Podati je potrebno zahteve po vzpostaviti testnih okolij pri naročniku s poudarkom na pripravo navodil za namestitvev okolij, čim višjo stopnjo avtomatizacije nameščanja, orodij za pripravo in hrambo testnih podatkov, itn
- Za potrebe osnovne simulacije prometnih scenarijev se podajao zahteve za simulacijsko okolje, kjer je potrebno predvideti, da se uporabi programsko opremo, namestitve in konfiguracijo zadnjega potrjenega testnega okolja v kombinaciji z orodjem za avtomatizirano pripravo/generiranje testnih podatkov. Predvideti je potrebno, da se simulacijsko okolje nadgrajuje skupaj z ostalimi okolji.
- **Varnostna testiranja:** Rešitev mora biti izdelana tako, da bo skladna in bo lahko prestala ustrezna varnostna testiranja, kjer so na ravni EU na voljo metodologije in orodja za izpostavljenost področje (npr. ENISA). Med drugim so na voljo tudi priporočila državam članicam EU glede zagotavljanja varnosti kritične infrastrukture, katere je potrebno v maksimalni meri upoštevati.
- **Testi samodejnega preklopa:** Posebno pozornost je potrebno posvetiti zahtevam vezanim na izvajanje testov zmogljivosti sistema za samodejno preklapljanje na sekundarni sistem v primeru izpada primarnega sistema. Predvideti je potrebno postopke izvedbe in glede na kritičnost sistema tudi periodiko izvajanja ter zahtevano dokumentacijo tovrstnega testiranja.

7. DOKUMENTACIJA IN PRENOS ZNANJA

Pri izdelavi dokumentacije za izvedbo je potrebno predvideti katero dokumentacijo mora izvajalec pri implementaciji in predaji informacijskih sistemov in programskih rešitev predati za uspešen zaključek projekta. Projektant/izvajalec mora glede na vsebino, obseg, kompleksnost in kritičnost IS ali programske rešitve predpisati, katera dokumentacija mora biti predana oziroma odložena v odlagališča naročnika in le to uskladiti z naročnikom. Dokumentacija mora biti skladna s standardi (kot na primer IEC/IEEE 82079-1-2019,...), dobrimi praksami in uporabljenimi metodologijami določenega življenjskega cika, zajemati pa mora vsaj:

Dokumentacijo za uporabniško programsko opremo, ki mora vsebovati:

- Krovni dokument uporabniške programske opreme s kratkim opisom vsebine, seznamom in lokacijo vse pripadajoče dokumentacije.
- Uporabniško dokumentacijo za vse nivoje uporabnikov, najmanj pa:
 - administratorje aplikacije,
 - uporabnike aplikacije
 - uporabi se princip vodenja uporabnika skozi posamezne korake, ki pripeljejo do rešitve nalog (primerno za nove uporabnike),
 - uporabi se princip, kjer so ukazi in postopki navedeni v obliki seznama (primerno za napredne uporabnike).
- Načrt testiranja, testne postopke, nabor testnih podatkov in poročila o testiranju.
- Seznam zunanjih orodij, ki niso del uporabniške programske opreme in so potrebna pri upravljanju ali razvoju oz. nadgradnjah uporabniške programske opreme.
- Dokumentacija izvedene analize rešitve. (Dokument sistemske analize).
- Dokumentacija o arhitekturi in zasnovi sistema.
- Podrobno tehnično dokumentacijo:
 - standardno dokumentacijo izvirne kode, kjer je ključno da vsebuje vse komentarje razvijalca, kar omogoča lažje branje in razumevanje kode,
 - dokumentacijo shem xml,
 - dokumentacijo vmesnikov spletnih storitev,
 - dokumentacijo programskih vmesnikov,
 - dokumentacijo uporabljenih lastnih ali tujih programskih komponent,
 - dokumentacijo postopkov in algoritmov, kar vključuje delovne tokove in vgrajena poslovna pravila,

- diagram odvisnosti med programskimi vmesniki in sistemi z analizo primernosti za virtualno okolje,
- navodila za namestitev v vsa okolja (npr.: testno, produkcijsko, šolsko, simulacijsko) z navedenimi predpostavkami, sistemskimi nastavitvami in omejitvami.

Dokumentacijo za strojno opremo, ki mora vsebovati:

- Izjavo proizvajalca ali zastopnika o mednarodni uveljavljenosti strojne opreme. (Uveljavljena v najmanj treh državah EU).
- Izjavo ponudnika strojne opreme glede podpore, garancije, pogarancijskih storitev (minimalna doba za vzdrževanje, nadomestne dele in priklopne aparate je tri leta), odzivnih časov v primeru okvare.
- Osnovna navodila v slovenskem jeziku.
- Dokumentacijo, ki dokazuje skladnost z:
 - nizkonapetostno direktivo (Direktiva 2014/35/EU),
 - direktivo o elektromagnetni združljivosti (Direktiva 2014/30/EU)
 - direktivo RoHS 3 (EU Directive 2015/863).

Dokumentacijo za sistemsko programsko opremo in prenos znanja:

Sistemska programska, ki oprema zajema:

- Operacijske sisteme (Linux, Windows, VMware,...).
- Gonilnike naprav (tiskalnik, mrežna kartica, video kartice, diskovna polja in pomnilne kapacitet, USB naprave, itn.).
- Komunikacijske programe.
- Sistemske (utility) programe (stiskanje datotek, urejanje datotek, protivirusni programi,...)
- Dokumentacijo o sistemskih nastavitvah za vse elemente sistema (podatkovna baza, aplikacijski strežnik, idr.) z opisom razlogov za spremembo privzete nastavitve

Dobavitelj sistemske programske opreme mora naročniku zagotoviti:

- Brezplačen dostop do baz znanja proizvajalca in spremljanje odprtih problemov preko spleta.
- Brezplačen spletni dostop do popravkov in nadgradenj (gonilniki, firmware) pri proizvajalcu opreme.
- Brezplačne storitve nadgradnje sistemske programske opreme v času garancije opreme v primeru funkcionalnih težav ali v primeru odstopanj od deklariranih lastnosti ponujene opreme.
- Dostop do tehnične podpore pri dobavitelju ali proizvajalcu.
- Pomoč pri reševanju tehničnih problemov v zvezi z nameščeno opremo v rednem delovnem času (v obdobju garancije).
- **Skladnost z zakonodajo:** Projektant mora v dokumentaciji za izvedbo dosledno in podrobno povzeti pridobljene podatke in informacije glede potreb in pričakovanj naročnika in le te opredeliti v funkcionalnih in tehničnih zahtevah. Zahteve morajo biti napisane transparentno, omogočati morajo jasno sliko pričakovanj naročnika, večinoma ne smejo biti zapisane tako da eksplicitno določajo tehnologijo le enega proizvajalca, razen v primerih, ko je to mogoče trdno argumentirati.
- **Prenos znanj:** Zunanji izvajalci za razvite in predane rešitve pripravijo zahtevano dokumentacijo ter prenesejo potrebno znanje za razumevanje delovanja in upravljanje rešitev na strokovnjake naročnika. Izobraževanje mora biti izvedeno tako za tehnično osebje, kot tudi za uporabnike.

8. PREDAJA SISTEMA/REŠITVE

Pri končni predaji aplikativne programske rešitve (oziroma sistema, ki predvideva tudi aplikativno programsko rešitev), v kolikor je bila le ta izdelana za DARS d.d. in ni licenčna, mora izvajalec naročniku predati vse potrebne elemente, s tem pa zagotoviti popolno naročnikovo neodvisnost pri nadaljnjem upravljanju, nadgradnji in vzdrževanju, med in po preteku garancijskega obdobja, brez dodatnih licenčnih stroškov. V praksi to pomeni, da mora

naročnik pridobiti vso dokumentacijo potrebno za nadaljnji neodvisen razvoj rešitve. Kot je opredeljeno že v dokumentu samem, mora biti poleg predane dokumentacije opredeljene v točki 1.6, predene predvsem:

- Izvorne kode rešitve z vsemi elementi, to zajema tudi morebitne elemente, ki jih je izvajalec kupil na trgu ali dodatno razvil in so del rešitve, kot so določene knjižnice, gonilniki, kodirni in dekodirni kodeki,... (tu mora biti izvajalec že v projektni nalogi ali v PZI zavezan, **da je ta del rešitve last naročnika in bo ob prevzemu tudi predan brez dodatnih stroškov za le tega**)
- Administratorska in uporabniška navodila za celotno upravljanje in uporabo rešitve
- Vsa uporabniška imena in gesla, ki so v trenutku predaje aktivna v sami rešitvi ter njihove vloge
- **Namestitveni paket**, v katerem so opredeljeni natančni postopki namestitve rešitve ali sistema, na osnovi katerega lahko naročnik izvede namestitev samostojno in rešitev ali sistem vzpostavi do stanja popolnega funkcionalnega delovanja.
- V primeru aplikativne rešitve ali IS, ki je delno ali v celoti licenčen pa se poleg predhodno podanih zahtev predajo vse licence, ki so kakorkoli povezane s celotnim življenjskim ciklom in delovanjem le te, od načrtovanja, izdelave, implementacije, vzdrževanja in odstranitve. V primeru, da je izvajalec moral nabaviti tovrstne licence ali elemente rešitve, so le te last naročnika in morajo biti predvidene in predane naročniku, potreba po nabavi pa predhodno, v fazi priprave tehnološkega elaborata ali pa fazi razvoja usklajena in odobrena s strani naročnika.
- Celotna rešitev mora v celoti omogočati, da naročnik brez nobene omejitve dostopa z administratorskimi pravicami do vseh delov rešitve. Administratorski profili morajo enolično opredeliti kdo je uporabnik, ki je dostopal in izvajal spremembe v sistemu (npr. Admin-JNovak). Osrednji administratorski profil je predan v zapečateni ovojnici in ni predviden za uporabo v sistemu, razen ko pride do izrednih okoliščin, katere opredeli naročnik interno. Sistem mora izvajati zbiranje dnevniških zapisov vseh ključnih sprememb, ki jih administratorji lahko izvajajo.
- Protokol predaje rešitve ali IS predpiše naročnik.

Z vso prevzeto dokumentacijo in elementi rešitve naročnik prosto razpolaga v okvirih svojih poslovnih potreb.