

ZAHTJEVI · ARHITEKTURA · RAZVOJ · IZDANJE · RAD SUSTAVA

Od zahtjeva do kontroliranog izdanja.

Kako je sigurnost integrirana kroz cijeli životni ciklus razvoja softvera.

ID DOKUMENTA	VERZIJA	KLASIFIKACIJA	VLASNIK DOKUMENTA	CIKLUS REVIZIJE	OBJAVLJENO
ITO-TRUST-SDL-001	1.0	JAVNO	Inženjering	Godišnje	Listopad 2026.

Siguran softver počinje razumijevanjem

Siguran softver ne nastaje samo sigurnosnim testiranjem.

Sigurnost počinje razumijevanjem onoga što sustav treba raditi, koje informacije obrađuje, tko će ga koristiti, s kojim sustavima komunicira i što bi se moglo dogoditi ako mu se ugrozi povjerljivost, cjelovitost ili dostupnost.

ITO stoga integrira sigurnosna razmatranja kroz cijeli životni ciklus razvoja softvera. Naš pristup kombinira analizu zahtjeva, arhitekturu, prakse sigurnog razvoja, pregled, testiranje, kontroliranu implementaciju i kontinuirano održavanje.

Ovaj dokument daje pregled tog procesa na visokoj razini.

SIGURNOST POČINJE PITANJIMA

Što sustav treba raditi?

Koje informacije obrađuje?

Tko će ga koristiti?

S kojim sustavima komunicira?

Što ako se ugrozi C, I ili A?



Ovaj dokument namjerno ne otkriva osjetljive interne procedure, sigurnosne konfiguracije ni pojednosti izvedbe specifične za proizvod.

SADRŽAJ

01	Svrha	02
02	Naš inženjerski životni ciklus	03
03	Zahtjevi i rizik	04
04	Sigurna arhitektura	05
05	Siguran razvoj	06
06	Pregled koda i kontrola promjena	07
07	Sigurnosna verifikacija	07
08	Ovisnosti o trećim stranama	08
09	Kontrolirano izdanje i implementacija	08
10	Sigurnost nakon izdanja	09
11	Kontinuirano poboljšanje	10
12	Podijeljena odgovornost	10
13	Jamstva o razvoju	11
14	Naše inženjersko načelo	11

KONTROLA DOKUMENTA

ID DOKUMENTA	ITO-TRUST-SDL-001
VERZIJA	1.0
KLASIFIKACIJA	JAVNO
VLASNIK	Inženjering
CIKLUS REVIZIJE	Godišnje
OBJAVLJENO	Listopad 2026.

Osam faza, jedan kontinuirani ciklus



Sigurnost nije završna faza testiranja.
To je **inženjersko razmatranje** kroz cijeli
životni ciklus.

Razumjeti prije izgradnje

Siguran razvoj počinje prije pisanja izvornog koda.

ITO inženjerski timovi najprije nastoje razumjeti operativno okruženje u kojem će softver raditi. Relevantna razmatranja mogu uključivati:

Korisnici?

Tko će pristupati sustavu i s kojim odgovornostima?

Informacije?

Koje će se informacije obrađivati i koliko su osjetljive?

Integracije?

Koji će interni i vanjski sustavi komunicirati s rješenjem?

Dostupnost?

Koje su operativne posljedice nedostupnosti?

Revizijski trag?

Koje aktivnosti moraju biti sljedive?

Infrastruktura?

Gdje i kako će rješenje raditi?

Upravljanje?

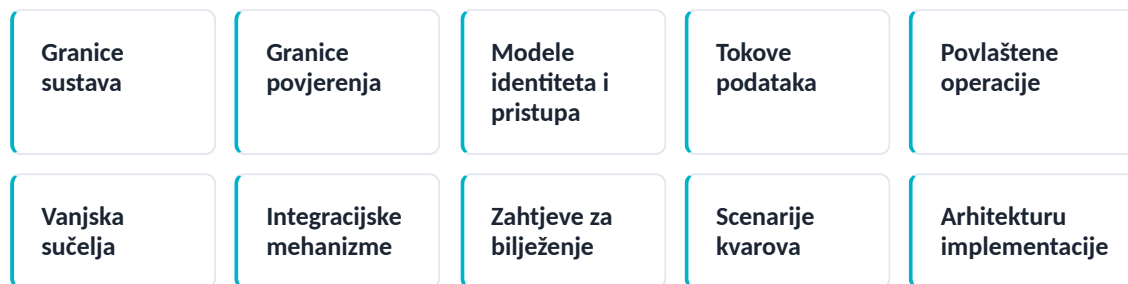
Koji institucionalni, ugovorni i regulatorni zahtjevi utječu na sustav?

Sigurnosni zahtjevi tada mogu postati dio zahtjeva rješenja, a ne dodatni sloj uveden nakon izrade.

Sigurnost počinje arhitekturom

Mnoga sigurnosna svojstva sustava određena su arhitektonskim odlukama donesenima prije početka izrade.

ARHITEKTURA ITO RJEŠENJA STOGA MOŽE RAZMATRATI



Cilj je **smanjiti nepotrebnu izloženost** i uspostaviti jasne granice između korisnika, servisa, informacija i vanjskih sustava.

Višeslojna zaštita

ITO daje prednost slojevitoj sigurnosnoj arhitekturi umjesto oslanjanja na jedan sigurnosni mehanizam. Ovisno o rješenju, kontrole mogu postojati na razinama:



Zakazivanje ili kompromitacija jedne razine ne bi smjeli automatski omogućiti neograničen pristup širem sustavu.

Razvoj s ugrađenom sigurnošću

Tijekom izrade od razvojnih inženjera očekuje se primjena praksi sigurnog inženjeringa primjerenih tehnologiji i aplikaciji koja se razvija.

ONE MOGU UKLJUČIVATI

01 Validaciju ulaznih i izlaznih podataka

02 Odgovarajuću autentifikaciju i autorizaciju

03 Sigurno rukovanje osjetljivim informacijama

04 Kontrolirano rukovanje pogreškama

05 Zaštitu od uobičajenih aplikacijskih ranjivosti

06 Prakse sigurne konfiguracije

07 Upravljanje ovisnostima o trećim stranama

08 Odvajanje osjetljive konfiguracije od aplikacijskog koda

Relevantne smjernice OWASP-a mogu se koristiti kao referenca za rizike sigurnosti aplikacija i prakse sigurnog razvoja.

Tajne i konfiguracija

IZBJEGAVA SE

Vjerodajnice, API ključevi, certifikati i druga osjetljiva konfiguracija nepotrebno ugrađeni u izvorni kod aplikacije.

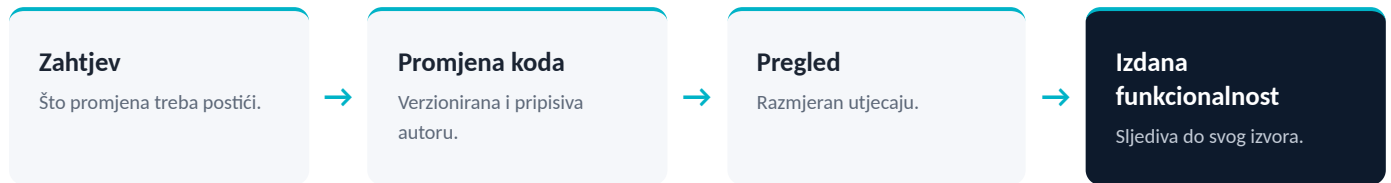
UMJESTO TOGA

Osjetljivom konfiguracijom treba rukovati u skladu s arhitekturom i modelom implementacije konkretnog rješenja.

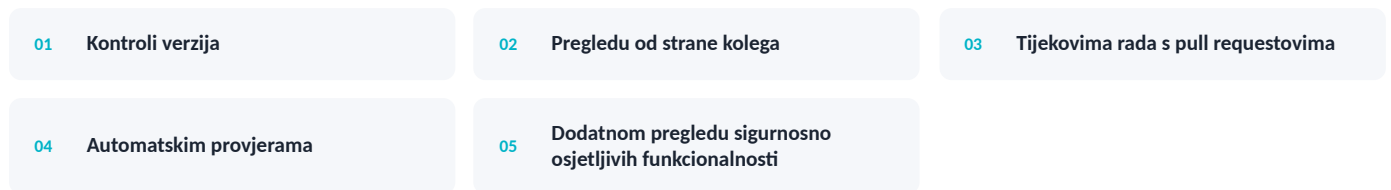
Promjene trebaju biti provjerljive i sljedive

Razvoj softvera zahtijeva kontrolirane promjene.

ITO razvojni tijekovi rada trebaju osigurati odgovarajuću sljedivost između softverskih zahtjeva, promjena izvornog koda i izdanih funkcionalnosti.



PROMJENE KODA MOGU PODLIJEGATI



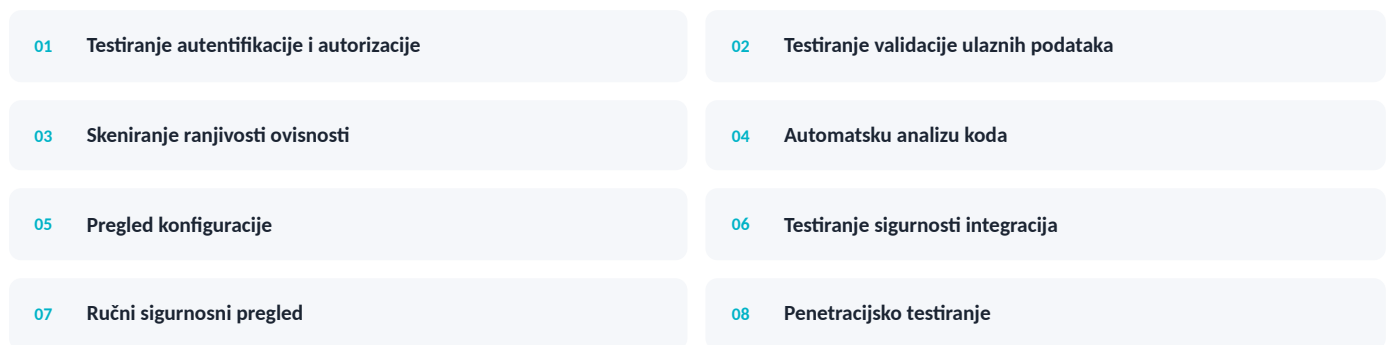
Razina pregleda ovisi o prirodi, složenosti i mogućem utjecaju promjene.

07 • SIGURNOSNA VERIFIKACIJA

Provjera prije izdanja

Sigurnosna verifikacija dopunjuje funkcionalno testiranje i testiranje kvalitete.

Ovisno o rješenju, izdanju i profilu rizika, aktivnosti verifikacije mogu uključivati:



DUBINA I KOMBINACIJA AKTIVNOSTI ODREĐUJU SE PREMA



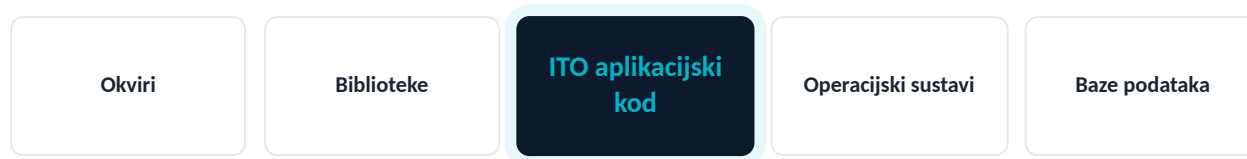
Suvremeni softver ima svoj lanac opskrbe

Suvremene aplikacije ovise o okvirima, bibliotekama, operacijskim sustavima, bazama podataka i drugim tehnologijama trećih strana.

Te ovisnosti mogu uvesti ranjivosti čak i kada se aplikacijski kod koji je razvio ITO nije promijenio.

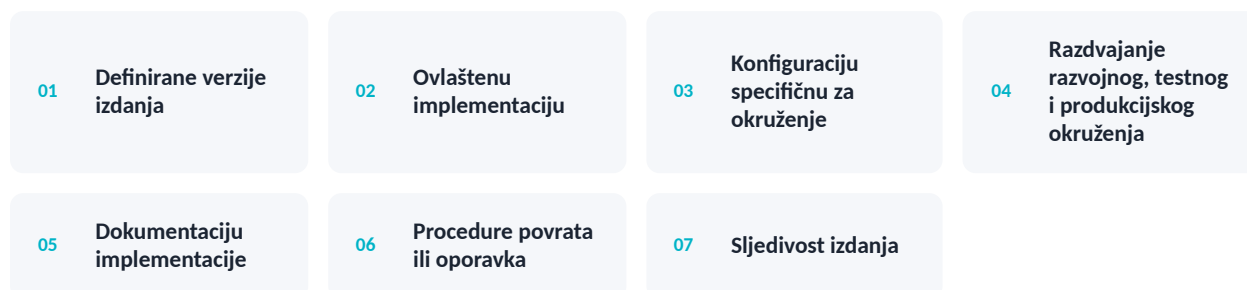
Relevantne komponente trećih strana stoga treba razmatrati tijekom razvoja i održavanja.

Kada se utvrde sigurnosne ranjivosti relevantnih ovisnosti, treba procijeniti njihovu primjenjivost i mogući utjecaj te razmotriti odgovarajuće otklanjanje.



Od izvornog koda do produkcije

Produkcijski softver treba uvoditi kroz kontrolirane procese izdanja i implementacije. Ovisno o projektnom modelu, kontrole mogu uključivati:



Produkcijska okruženja treba tretirati drukčije od razvojnih jer mogu sadržavati **operativne informacije** i podupirati **aktivne institucionalne procese**.

Sigurnost softvera nastavlja se u produkciji

Sigurno izdanje s vremenom može postati ranjivo.

Nove ranjivosti mogu se otkriti u aplikacijskim okvirima, ovisnostima, infrastrukturnim komponentama ili tehnologijama o kojima rješenje ovisi. ITO stoga održavanje tretira kao dio sigurnosnog životnog ciklusa. Aktivnosti mogu uključivati:

01

Procjena ranjivosti

Procjenjivanje ranjivosti relevantnih za podržani softver.

02

Ažuriranja ovisnosti

Ažuriranje pogođenih komponenti gdje je prikladno.

03

Sigurnosni ispravci

Razvoj korektivnih izdanja za potvrđene ranjivosti u komponentama koje održava ITO.

04

Operativne povratne informacije

Korištenje incidenata, nedostataka i povratnih informacija korisnika za poboljšanje budućih izdanja.

05

Upravljanje životnim ciklusom

Razmatranje statusa podrške i sigurnosnih implikacija temeljnih tehnologija.

Inženjerski procesi se razvijaju

Siguran razvoj nije statičan kontrolni popis. Inženjerske prakse moraju se razvijati kako se mijenjaju tehnologije, prijetnje, ranjivosti, regulatorna očekivanja i operativni zahtjevi. ITO stoga koristi iskustva stečena u razvoju, testiranju, produkcijskom radu, sigurnosnim procjenama i okruženjima korisnika kako bi s vremenom unapređivao svoje inženjerske prakse.

Životni ciklus sigurnog razvoja ne definira to što se ranjivosti nikada ne pronađu.

Definira ga sposobnost da ih se **otkrije, razumije, otkloni i iz njih uči.**

12 • PODIJELJENA ODGOVORNOST

ITO razvija i održava softver unutar ekosustava koji može uključivati i infrastrukturu kojom upravlja korisnik, vanjske sustave i tehnologije trećih strana. Siguran rad stoga zahtijeva koordinaciju između:

ITO

Arhitektura aplikacije, inženjering, sigurnost proizvoda i održavanje unutar dogovorenog opsega.

INSTITUCIJA KORISNIK

Korisnici, ovlasti, upravljanje, operativne procedure i infrastruktura pod kontrolom korisnika.

PRUŽATELJI INFRASTRUKTURE

Hosting, mreža, operacijski sustavi i infrastrukturne kontrole kada se njima upravlja odvojeno.

TEHNOLOŠKI PARTNERI

Sigurnost i životni ciklus samostalno pruženih proizvoda i usluga.

Detaljne odgovornosti ovise o konkretnoj implementaciji i ugovornom aranžmanu.

Informacije za tehničku dubinsku analizu

Korisnici i partneri mogu tijekom tehničke dubinske analize ili procjene dobavljača zatražiti dodatne informacije o ITO inženjerskim procesima.

Gdje je prikladno, ITO može pružiti dodatnu dokumentaciju o sigurnom razvoju, sigurnosnom testiranju, arhitekturi, upravljanju ranjivostima i procesima izdanja.

Osjetljive interne procedure i tehničke informacije mogu se, kada je opravdano, pružiti kroz kontrolirane kanale jamstava.

ZA DODATNE INFORMACIJE

[ITO sigurnosni whitepaper](#)

[ITO Politika odgovornog otkrivanja ranjivosti](#)

[ITO Privatnost i upravljanje podacima](#)

[ITO Načela odgovorne umjetne inteligencije](#)


14 • NAŠE INŽENJERSKO NAČELO

Sigurnost nije nešto što dodajemo gotovom softveru.

Ona utječe na to kako razumijemo zahtjeve, dizajniramo arhitekturu, pišemo kod, provjeravamo promjene, implementiramo sustave i održavamo ih tijekom vremena.

NAŠ CILJ

Izgraditi softver koji pri isporuci nije samo funkcionalan, već je i **održiv**, **revizijski provjerljiv** i **sposoban za siguran razvoj** kroz cijeli operativni životni ciklus.



ITO TRUST DOKUMENTI

Engineering Digital Trust.

POVEZANI DOKUMENTI

ITO-TRUST-SEC-001

ITO sigurnosni whitepaper

ITO-TRUST-PRV-001

ITO Privatnost i upravljanje podacima

ITO-TRUST-AI-001

ITO Načela odgovorne umjetne
inteligencije

DOKUMENT

ITO Pregled sigurnog razvoja softvera

ID • VERZIJA • KLASIFIKACIJA

ITO-TRUST-SDL-001 • 1.0 • JAVNO

VLASNIK • REVIZIJA

Inženjering • Godišnje • Listopad 2026.

KONTAKT

security@ito.dev • ito.dev/en/trust-center

TVRTKA

IT Odjel Mostar d.o.o., Mostar, Bosna i Hercegovina