

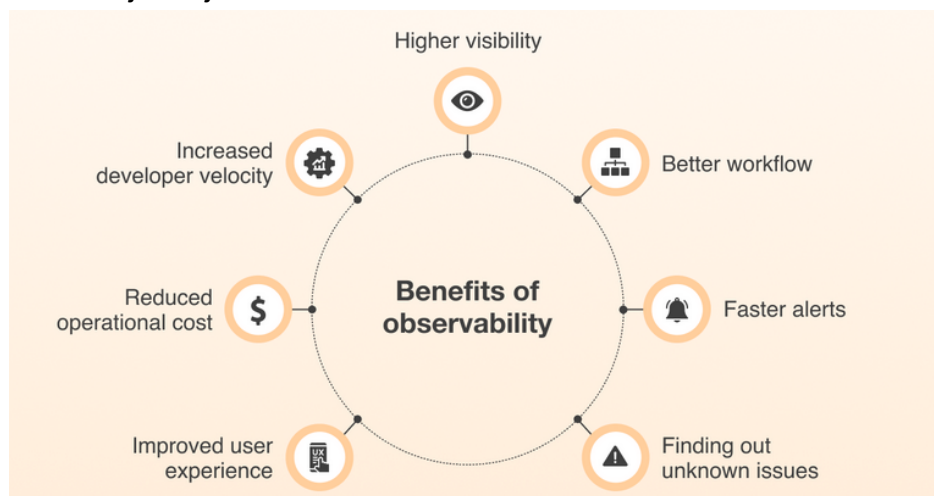
# Nadgledanje i monitoring sistema

Šta je opazivost ili nadgledanje (eng. *observability*)?

Softversko nadgledanje (eng. "Software observability") predstavlja praksu praćenja, merenja i razumevanja ponašanja softverskog sistema u realnom vremenu. Uključuje sakupljanje i analiziranje podataka o performansama, dostupnosti i pouzdanosti softverskih sistema radi brzog otkrivanja, dijagnostikovanja i rešavanja problema. Cilj softverskog nadgledanje je poboljšanje kvaliteta softvera i povećanje njegove pouzdanosti i dostupnosti tako što će programerima i timovima za održavanje biti omogućeno da imaju uvid u to kako softver funkcioniše u produkcioniom okruženju.

Da bi postigli softversko nadgledanje, organizacije koriste kombinaciju alata i tehnika kao što su beleženje događaja (eng. "logging"), merenja (eng. "metrics"), praćenje (eng. "tracing") i slanje upozorenja (eng. "alerting"). Beleženje događaja podrazumeva sakupljanje i skladištenje podataka o događajima i aktivnostima unutar softverskog sistema, kao što su interakcije korisnika, poruke o greškama i sistemski događaji. Merenja pružaju numeričke podatke kojima se može izmeriti performansa i zdravlje softverskog sistema, poput vremena odziva, propusnosti i stopa grešaka. Praćenje uključuje praćenje toka zahteva i poruka kroz softverski sistem, što programerima i timovima za održavanje omogućava da razumeju kako sistem funkcioniše u različitim fazama obrade. Na kraju, slanje upozorenja omogućava timovima da dobiju obaveštenja kada se detektuju problemi ili anomalije, što im omogućava da brzo reaguju i rešavaju probleme pre nego što utiču na korisnike.

Softversko nadgledanje je neophodna za moderne prakse razvoja i održavanja softvera, posebno za arhitekture zasnovane na mikroservisima, koje mogu biti složene i distribuirane preko više sistema i okruženja. Softversko nadgledanje omogućava timovima da identifikuju i rešavaju probleme brzo, poboljšaju kvalitet softvera i pružaju bolja korisnička iskustva. Takođe, omogućava timovima da identifikuju trendove i obrasce tokom vremena, što im pomaže da donose odluke zasnovane na podacima o tome kako optimizovati i unaprediti softverske sisteme. Ukratko, softversko nadgledanje je kritična praksa za organizacije koje žele da isporuče dobra rešenja svojim korisnicima.



## Šta je monitoring?

Monitoring (eng. "Software monitoring") predstavlja praksu kontinuiranog posmatranja i merenja ponašanja i performansi softverskog sistema u realnom vremenu. Uključuje sakupljanje i analizu podataka o dostupnosti, pouzdanosti i performansama softverskih sistema radi brzog otkrivanja i dijagnostikovanja problema. Cilj softverskog monitoringa je obezbediti optimalno funkcionisanje softverskog sistema i identifikovati probleme pre nego što utiču na korisnike ili poslovne operacije.

## Razlika između monitoringa i nadgledanja?

Iako su povezani, ne predstavljaju skroz dva ista koncepta. Kroz monitoring najčešće postoji prekonfigurisan grafički interfejs koji je tu da se vidi kako se sistem ponaša kroz vreme. Njegova uloga je da nam pokaže sve anomalije i sve otkaze koje smo imali u sistemu. Osnovna pretpostavka monitoringa je da ste u stanju da predvidite sve probleme koje očekujete da vidite u sistemu, a to je jako često nemoguće međutim nije daleko od nečeg što je potencijalno izvodljivo.

U sistemu nadgledanja najčešće postoje neki alati koji nude opciju pretraživanja metrika, logova kao i traceova i samim tim možete brzo doći do informacija koje vas zanimaju.

## Logovi

Logging je proces koji podrazumeva zapisivanje informacija o događajima i akcijama koje se dešavaju u okviru softverskog sistema. Ove informacije se obično skladište u log fajlovima ili u centralizovanim bazama za logove, koji se kasnije mogu analizirati kako bi se razumeo način funkcionisanja sistema i identifikovali problemi ili greške koje su se možda javile. Logovanje je važan alat za programere i IT stručnjake jer pruža detaljan zapis aktivnosti sistema koji se može koristiti za otklanjanje problema, debugovanje i optimizaciju performansi.

Logovi obično sadrže informacije o interakcijama korisnika, sistemskim događajima, porukama o greškama i drugim relevantnim podacima. Mogu takođe sadržati i dodatni kontekst, kao što su datum i vreme događaja, korisnik koji ga je pokrenuo i specifični komponenti sistema koji su bili uključeni. Ove informacije se mogu koristiti za identifikovanje obrazaca ponašanja, praćenje specifičnih problema i dijagnostikovanje problema koji bi inače bili teško uočljivi. Treba voditi računa o tome šta se zapisuje u ove stvari a šta ne.

Logovanje se može primeniti na različitim nivoima softverskog sistema, od nivoa aplikacije do nivoa operativnog sistema. Različiti okviri i biblioteke za logovanje postoje za različite programske jezike i platforme, a ovi alati se mogu prilagoditi za hvatanje određenih tipova podataka ili zapisivanje događaja u određenom formatu.

Iako je logovanje moćan alat, može takođe generisati veliku količinu podataka tokom vremena. Važno je pažljivo upravljati log fajlovima kako bi se sprečilo da zauzimaju preveliki prostor za skladištenje ili da utiču na performanse sistema. Ovo se može uraditi implementiranjem politika rotacije logova, postavljanjem odgovarajućih nivoa beleženja i redovnom analizom i uklanjanjem nepotrebnih podataka iz log fajlova.

## Metrike

Metrike su kvantitativna merenja koja se koriste za procenu kvaliteta i performansi softverskog sistema. One se koriste kako bi se dobio uvid u različite aspekte procesa razvoja softvera, kao što su dizajn, kvalitet koda, testiranje i održavanje. Sakupljanjem i analizom metrika, softverski programeri i menadžeri projekta mogu identifikovati oblasti koje treba poboljšati i pratiti napredak ka postizanju ciljeva razvoja.

Neke od često korišćenih metrika observabilnosti su:

- Latencija: meri vreme potrebno da se zahtev obradi u sistemu
- Stopa grešaka: meri učestalost grešaka ili izuzetaka koji se javljaju u sistemu
- Protok: meri količinu posla koji sistem može da obavi u datom periodu vremena
- Zasićenost: meri nivo iskorišćenja resursa u sistemu, kao što je korišćenje CPU-a ili memorije
- Kapacitet: meri maksimalnu količinu posla koji sistem može da obradi pre nego što postane preopterećen
- Mnoge druge

Pored praćenja metrika observabilnosti, takođe je važno uspostaviti pragove ili osnove za ove metrike, kako bi se detektovalo kada one premaše očekivane nivoe. To se može učiniti korišćenjem sistema za upozorenje i obaveštenja, koji mogu obavestavati razvojne timove kada metrike dostignu kritične nivoe ili kada se u sistemu dogode određeni događaji.

## Trace zapisi

Tracing uključuje snimanje događaja koji se javljaju u softverskom sistemu, a zatim analizu dobijenih podataka u cilju dobijanja uvida u to kako sistem funkcioniše. Tracing se može koristiti za dijagnosticiranje širokog spektra problema, od problema s performansama do grešaka i padova sistema. Praćenjem izvršavanja softverskog sistema, programeri mogu identifikovati uska grla, locirati problematičan kod i bolje razumeti način na koji sistem funkcioniše u realnom vremenu.

Tracing se takođe može koristiti za identifikovanje potencijalnih sigurnosnih ranjivosti u softverskim sistemima. Analizom podataka dobijenih pomoću tracing-a, programeri mogu identifikovati sumnjivo ponašanje koje može ukazivati na sigurnosni problem i preduzeti korake kako bi se te ranjivosti otklonile prije nego što ih napadači iskoriste.

## Alerting

Alerting se odnosi na proces slanja obaveštenja ili upozorenja sistemskim administratorima ili drugom relevantnom osoblju kada se određeni događaji ili uslovi pojave u softverskom sistemu. Ova upozorenja mogu obavestiti odgovarajuće osoblje o problemima koji zahtevaju hitnu pažnju, omogućavajući im da preduzmu akciju kako bi rešili problem pre nego što postane ozbiljniji. Može biti pokrenuto iz različitih događaja ili uslova, kao što su neuspesi sistema, greške, visoka upotreba resursa ili bezbednosne pretnje. Upozorenja se mogu dostavljati putem različitih kanala, uključujući e-poštu, SMS ili putem kontrolne table ili druge monitoringске platforme.

Jedna od glavnih prednosti softverskog upozoravanja je da omogućava proaktivno praćenje softverskog sistema. Umesto da se oslanja na sistemskog administratora ili drugo osoblje da stalno prati sistem, upozorenja se mogu postaviti kako bi ih obavestili o specifičnim događajima ili uslovima koji zahtevaju njihovu pažnju. To može pomoći da se spreče problemi da eskaliraju u ozbiljnije probleme koji bi mogli uticati na performanse, pouzdanost ili sigurnost sistema. Još jedna prednost softverskog upozoravanja je da može pomoći da se osigura da odgovarajuće osoblje bude obavešteno o problemima u pravo vreme. Pružanjem obaveštenja čim se pojavi problem, sistemski administratori mogu odmah preduzeti akciju kako bi rešili problem i minimizirali njegov uticaj. Ovo može pomoći u sprečavanju vremena neaktivnosti, gubitka podataka ili drugih negativnih posledica koje bi mogle nastati usled neuspeha sistema.