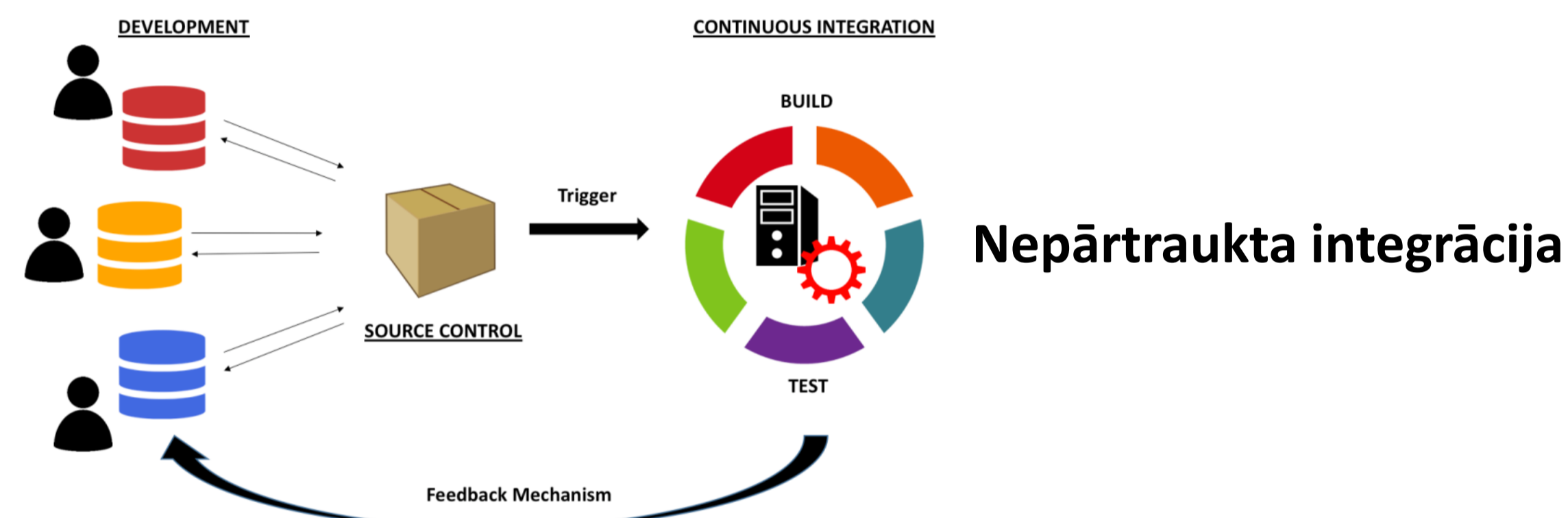


## PROBLĒMA

Uzņēmumam ir nepieciešamība saīsināt programmatūras izstrādes dzīves ciklu, lai spētu ātrāk piegādāt produktu, kā arī uzlabot komunikāciju un darba kultūru gan izstrādes komandā, gan ārpus tās – sadarbībā ar pasūtītāju.

## IZSTRĀDDARBINĀŠANAS PRINCIPI



## REZULTĀTI

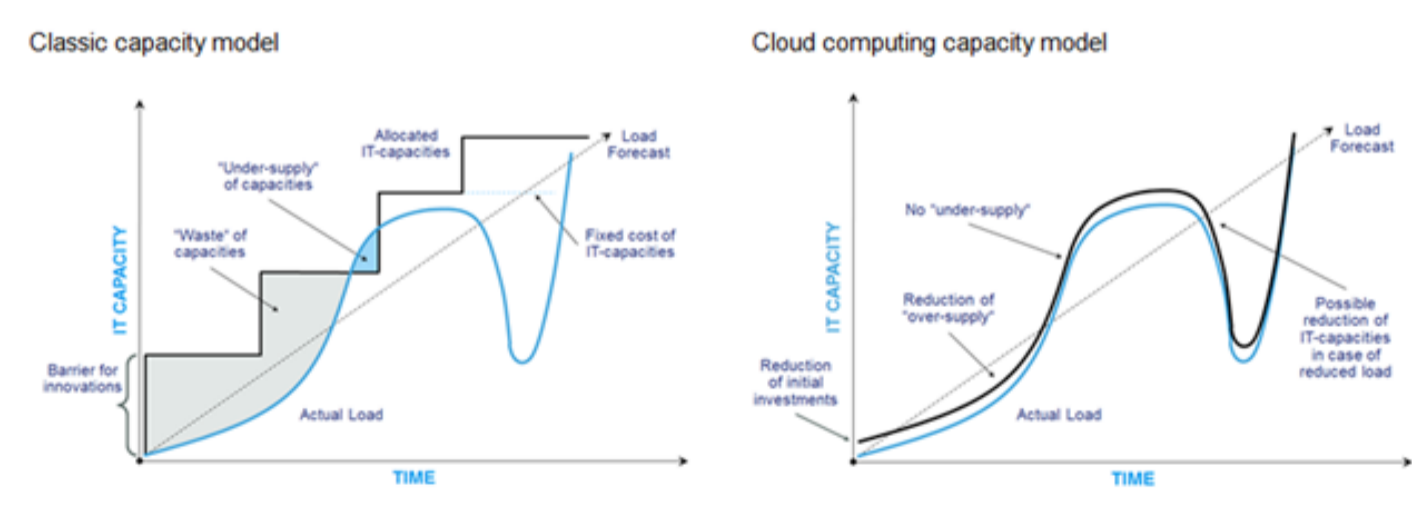
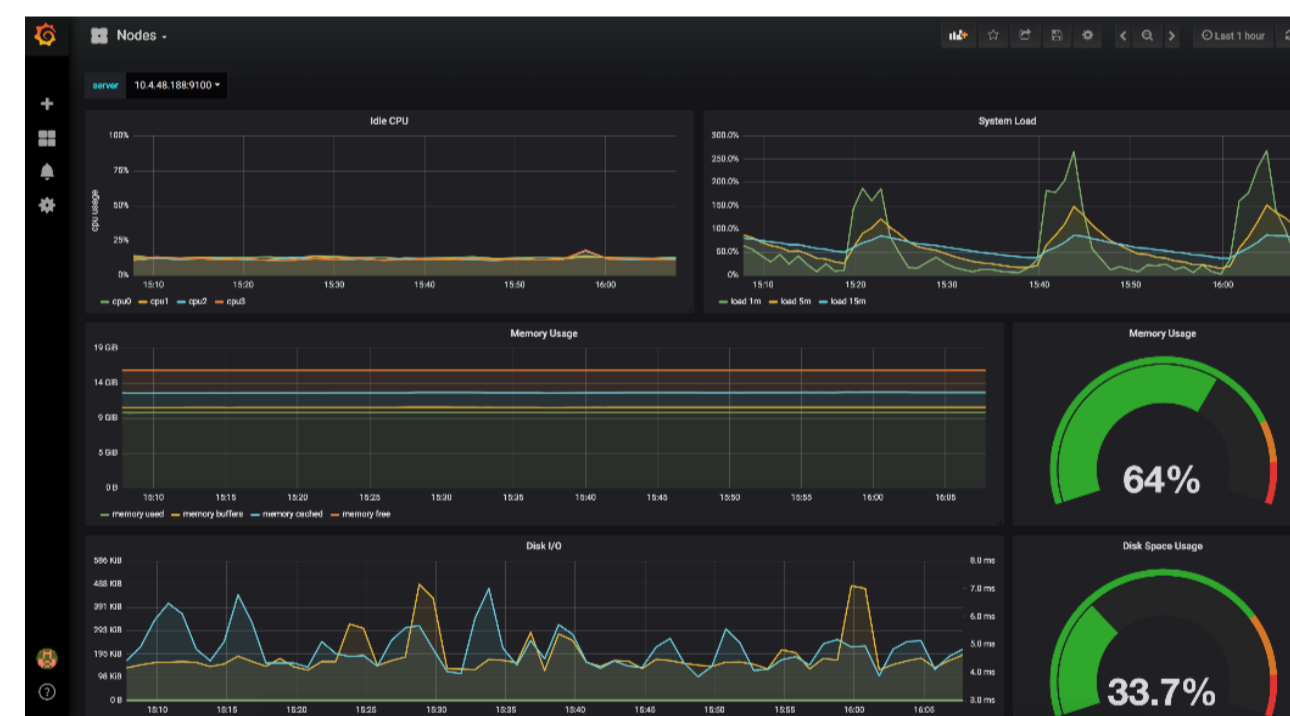
Apzinātas un apraktītas programmatūras izstrādes metodoloģijas un ietvari, kas uzskatāmi par izstrāddarbināšanas priekštecēm – ūdenskrituma modelis, ātrā lietojumprogrammu izstrāde (RAD), *Scrum*, ekstrēmā programmēšana (XP), spējā izstrāde.

Aprakstīti un apzināti izstrāddarbināšanas principi un rīki, piemēram, nepārtraukta integrācija un piegāde un uzraudzība, haosa inženierija un antitrauslums, mikropakalpju arhitektūra, infrastruktūra kā kods, u.c.

## MĒRĶIS

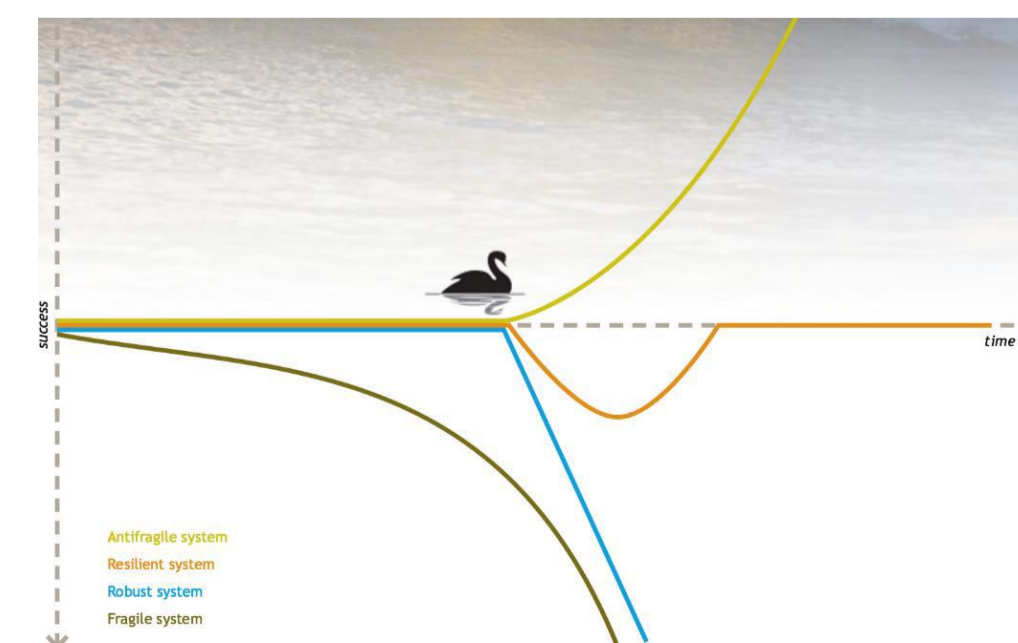
Apzināt un apraktīt izstrāddarbināšanas vēsturi, rašanās iemeslus, principus un rīkus. Pēc tam uzņēmumā novērtēt esošo situāciju un mēģināt šīs zināšanas pielietot, lai optimizētu uzņēmuma biznesa un izstrādes procesus, novērtēt ieguvumus.

### Nepārtraukta uzraudzība



### Mākoņskaitļošana un Automātiska mērogošana

### Antitrauslums (antifragility)



## SECINĀJUMI

1. Programmatūras izstrāde sākotnēji bija lineāra, bet vēlāk kļuva iteratīva. Šo pārmaiņu galvenais dzinējs bija nepieciešamība reaģēt uz izmaiņām prasībās.
2. Radās vairākas metodoloģijas, kas ieviesa dažādas vērtības un principus, kas atrisināja problēmas, kuras bija sastopamas ūdenskrituma modelī.
3. Izstrāddarbināšana radās kā atbilde nepieciešamībai ieviest spējās izstrādes vērtības praksē un šķelšanās starp programmatūras izstrādātājiem un infrastruktūras administratoriem dēļ.