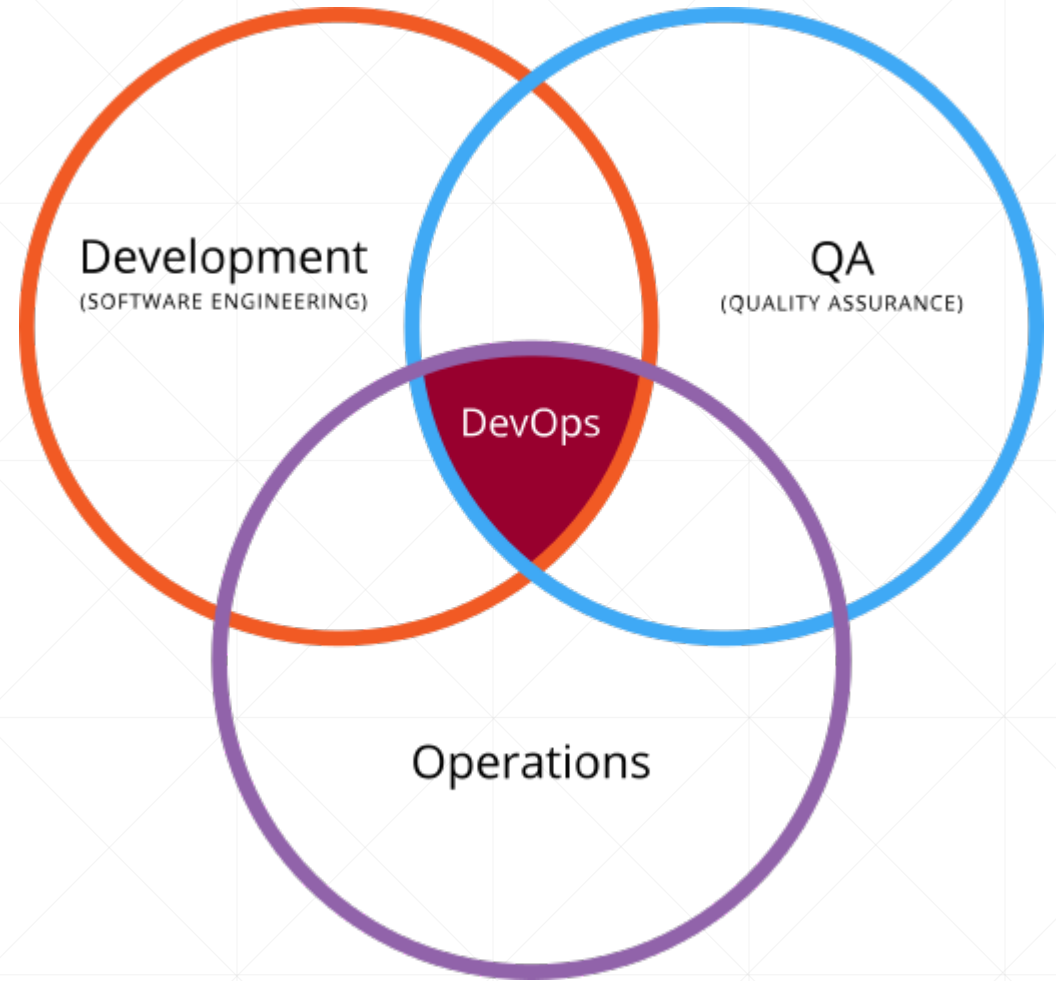


Ohjelmistotuotanto 2

DevOps ja CI/CD

Mitä on DevOps?

- DevOps = "a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality"
- DevOps on joukko menetelmiä, jotka saumattomasti yhdistävät softan kehittämisen ja käyttöönoton laadukkaasti (oma suomennos)



CI/CD pipeline

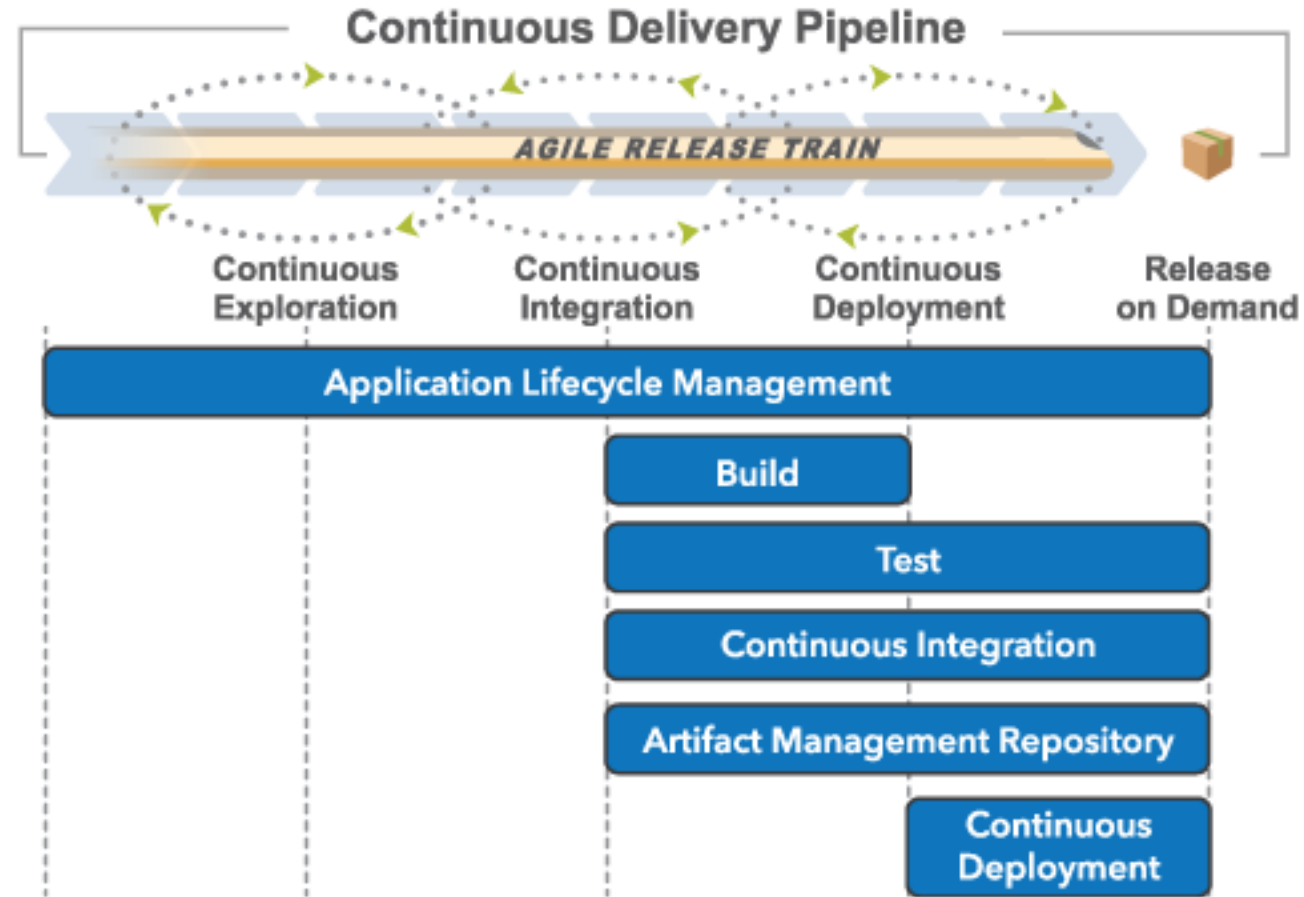
- Nykyään DevOpsin sijaan puhutaan monesti CI/CD-pipelinesta, jossa
 - CI = continuous integration = tuote pidetään joka hetki jakelukelpoisena
 - CD = continuous delivery = tuotteen jakelu eli tuotantokäyttöön siirtäminen on automatisoitu
 - CI
 - käytetään versionhallintaa
 - tuotteen kehitysversiot pidetään erillään tuotantoversioista
 - testaukset automatisoidaan, jotta tuotemuutokset on helpompi testata
 - myös muu automatisoitu laadunvalvonta, analysointi, mittaukset, monitorointi
-

DevOps vs. perinteinen vesiputousmalli

- Vesiputousmallin mukaan ohjelmiston kehittäminen etenee suunnittelusta ja toteutuksesta testauksen kautta tuotantoon
 - Kehittäjien ja operoijien välillä on muuri, tieto ei liiku tai se liikkuu hitaasti
 - mitkä ovat tuotantokäytön todelliset ongelmat?
 - miten tuotteeseen voisi saada nämä muutokset?
 - kaikki muutokset ovat iso työ, koska koko ketju pitää käydä läpi kokonaisuudessaan
 - DevOps: poistetaan tämä muuri, rakennetaan jatkuvan parantamisen periaatteeseen nojautuva, yhteensopivilla työkaluilla toteutettu saumaton kehittämisketju
-

Jatkuvan toimituksen hallinta

- DevOps-menettelytavat ja työkalut mahdollistavat ohjelmistotuotteiden joustavan ja laadukkaan ylläpitämisen ja muutokset
- Kun ohjelmiston toiminnassa mitataan poikkeama tai huomataan muutostarve, muutokset voidaan tehdä helposti, testata automatisoidusti ja toimittaa uusi versio tuotantoon automatisoidusti



DevOps-filosofia tiivistetysti

- Yhteistyö, automatisointi, jatkuva parantaminen, mittaaminen, jakaminen
- (lähde <https://www.atlassian.com/devops>)

The CALMS Framework for DevOps



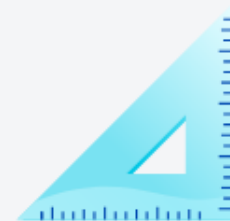
Culture



Automation



Lean



Measurement



Sharing

DevOps työkalut

Yhteistyön työkalut: Jira, Slack, Teams, Trello,---

Jatkuva kehittäminen Plan – muutosten suunnittelu

- Code –
- koodin kehitysympäristöt,
 - lähdekoodin hallinta, versionhallinta

Build

Jatkuva testaaminen Test – jatkuvan testauksen työkalut

Jatkuva integraatio Release - infrastruktuurin hallinta

Jatkuva jakelu

- Deploy –
- konfiguraation hallinta, infrastruktuuri koodina (IaS)
 - jakelupaketointi

Operate – tuotantokäyttö

Jatkuva seuranta Monitor –tuotantokäytön valvonta, palvelulaadun varmistus

- Eclipse, VSCode...
- Git, SVN, JIRA...

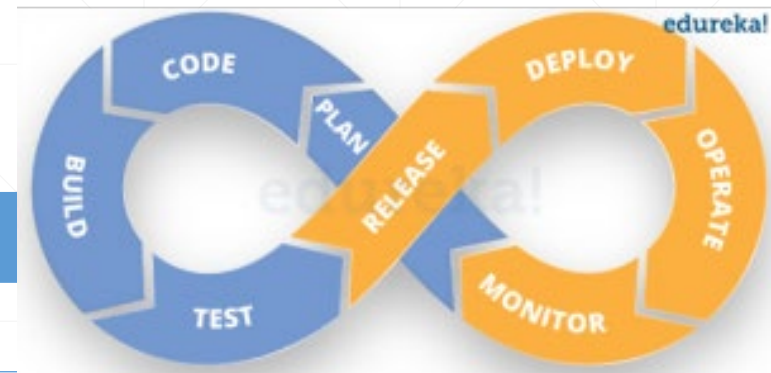
Apache Maven, Apache Ant, Gradle

Selenium, TestNG, JUnit/ NUnit

Jenkins, Hudson, Bamboo

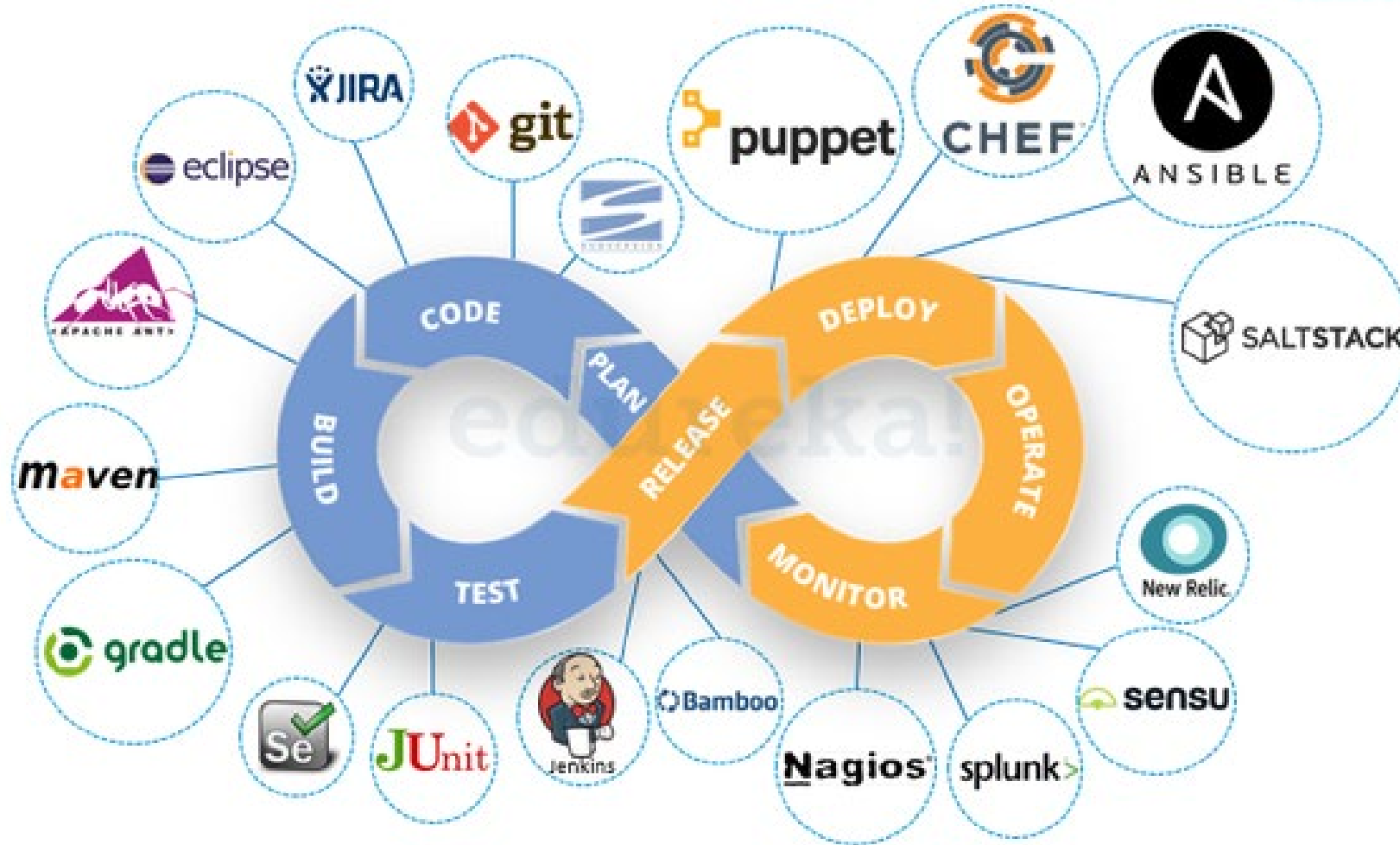
- Puppet, Chef, Ansible, and SaltStack
- Docker

Splunk, ELK Stack, Nagios, Sensu, NewRelic



Esimerkkejä DevOps-työkaluista (v.2018)

edureka!



Tärkeimmät työkalut

Erityisen tärkeitä työkaluja ovat:

- Versionhallinta (Git ym) kehitys- ja tuotantoversioiden ylläpitoon
 - Testaussovellukset automatisoituun testaamiseen
 - Integrointialusta (esim. Jenkins) joka pystyy ajamaan eri DevOps-ketjuun kuuluvia ohjelmia automatisoidusti (testaus, jakelupaketointi jne)
 - Infrastruktuuri koodina-ohjelmat (esim. Puppet), eli ohjelmat joille käyttöjärjestelmäkoonpanojen konfigurointi voidaan välittää kuvaustiedostoina
-