

Adoptarea unei culturi DevOps

Ioana Negoită

Informatică pentru Managementul Afacerilor, Universitatea Româno-Americană

Abstract

DevOps este unul dintre studiile conceptuale pentru integrarea revizuirii operaționale și de dezvoltare pentru infrastructură și sistem informațional. Companiile moderne de dezvoltare se confruntă cu provocări multiple pentru a răspunde cerinței de a dezvolta un sistem software și a menține calitatea software-ului. DevOps este o abordare care îmbină rolurile tradiționale ale software-ului și lucrează pe o modalitate de a îmbunătăți comunicarea pentru a perfecționa rata de frecvență a implementării și a menține calitatea software-ului. Lucrarea de față arată cum DevOps a fost introdus ca un set de reguli, instrumente și practici pentru a crea modalități mai eficiente de a face față provocărilor viitoare și existente legate de dezvoltarea software-ului și de a-și menține scopul cu avantajele sale operaționale. DevOps este de fapt un model care combină practici și instrumente care ajută la creșterea valorilor de bază.

Cuvinte cheie: DevOps, Dezvoltare continuă, integrare continuă, implementare continuă.

Introducere

Cererea pentru persoanele cu abilități DevOps crește rapid peste tot în lume, deoarece companiile obțin rezultate foarte bune din partea DevOps. Organizațiile care folosesc practicile DevOps au o funcționare foarte ridicată: implementează coduri de până la 30 de ori mai frecvent decât concurenții lor, acesta din urmă având o rată de implementare de succes de doar 50%, conform raportului State of DevOps 2013-2019 (Liu, 2019).

Toate aceste ne îndrumă să credem că există o mulțime de ingineri DevOps pe piața muncii, însă, doar 18 la sută dintre organizații dețin de fapt ingineri DevOps. Acest lucru se datorează, în parte, faptului că ceea ce fac inginerii DevOps este într-un continuu flux de implementare și mișcare. Totuși, între ianuarie 2017 și ianuarie 2018, listările pentru joburile DevOps de pe True.com au crescut cu 75%. Pe LinkedIn.com, mențiunile despre abilitățile DevOps necesare la locul de muncă au crescut cu 50 la sută în aceeași perioadă (Earnshaw, 2013).

Ce este DevOps?

Un inginer DevOps este un profesionist IT care lucrează cu dezvoltatorii de software, operatorii de sistem (SysOps) și alt personal IT de producție pentru a supraveghea lansările de cod și implementarea acestora. Rolul solicită pe cineva care are abilități relevante pentru a depăși barierele tradiționale dintre dezvoltarea de software, QA („quality assurance” - asigurarea calității) echipele de testare și operațiuni IT, și să încurajeze un mediu colaborativ, holistic (David, 2014, pp. 1-20).

DevOps este centrat în jurul următoarelor obiective:

- Construiește o cultură a efortului coordonat, care separă depozitele convenționale și măsurile de bază de realizare.
- Menține calitatea actuală în timp ce creează îmbunătățiri pentru viața produsului prin utilizarea DevOps.
- Codificare sau script
- Re-inginerie de proces

Cum au apărut inginerii DevOps?

Totul a început cu primul model de ciclu de viață al software-ului, modelul cascadă (Figura 1).

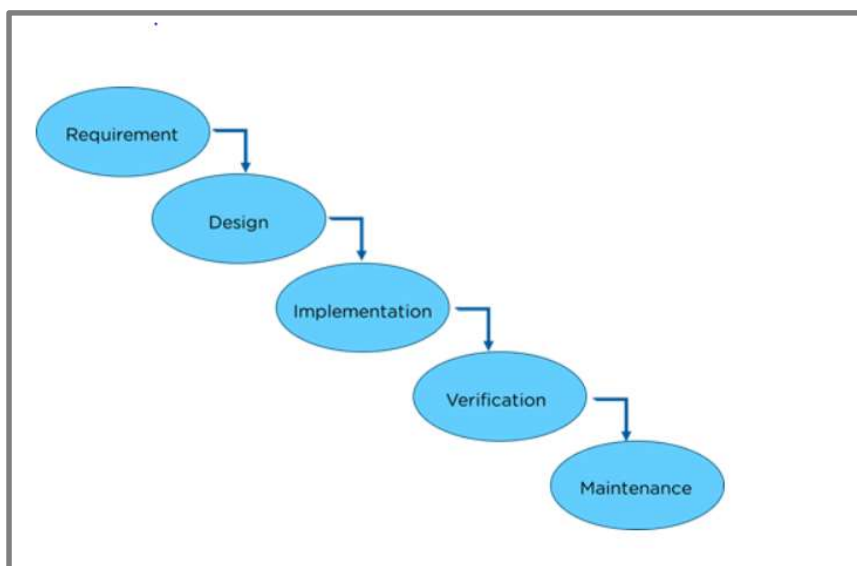


Figura1. Ciclu de viață al software-ului (McNeill, 2019)

Diagrama de mai sus arată fiecare etapă a modelului cascadei. Problema în cazul acestui model a fost lipsa de comunicare între clientul care oferă cerințele pentru produsul software dorit și echipa care-l dezvoltă. Acest lucru a dus adesea la proiectarea greșită a produsul software și la pierderea de timp și resurse.

În continuare, a venit metodologia agilă, care a introdus o buclă de feedback continuă între stadiul cerințelor și stadiul de dezvoltare (Figura 2), fără a avea, însă, nicio prevedere pentru întreținere.

Cu DevOps, există comunicare între echipa de dezvoltare și echipa implicată în întreținere și testare (echipa operațională), de aici și numele „Dev-Ops”.

DevOps este o abordare care se concentrează pe reunirea echipei de dezvoltare și operațiuni pentru a obține produse și servicii cu eficiență și calitate maximă. DevOps inițiază automatizarea pe cât posibil folosind mai multe instrumente.

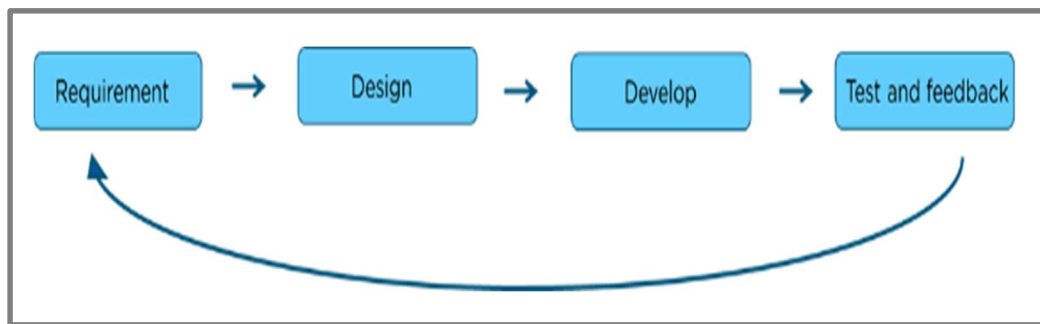


Figura 2. Etapele modelului cascadă (Nair, 2019)

Funcționează DevOps?

La fel ca și modelul cascadei și metodologia agila, DevOps are o serie de etape (Figura 3), prezentate succint în continuare.

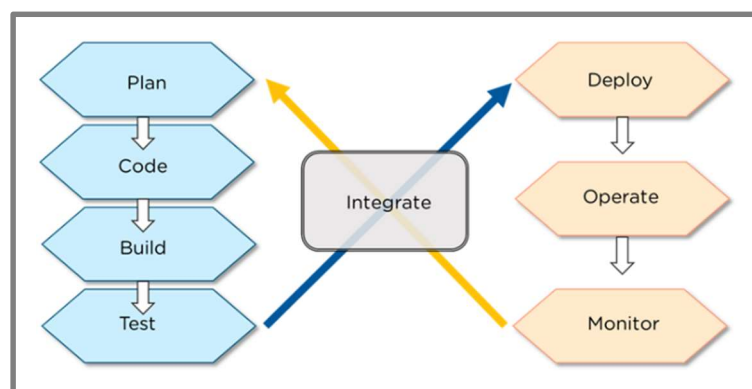


Figura. 3 Etapele fluxului DevOps (Moy, 2019)

(1) Planul și codul. Strategia și codul sunt dezvoltate în această etapă. Instrumentele utilizate aici sunt axate pe controlul versiunilor și oferă un depozit în acest scop.

(2) Construire. Etapa de construire este preocupată de preluarea automată a codului sursă din depozite și de ambalarea acestora în aplicații executabile.

(3) Testare. Testarea în DevOps este automatizată în conducta de compilare pentru a se asigura că aplicațiile sunt implementabile.

(4) Promovare. Pe măsură ce aplicațiile sunt gata, acestea sunt trimise la stadiul de implementare și sunt configurate continuu la starea corectă.

(5) Monitorizare. Produsele lansate sunt monitorizate continuu și feedback-ul este utilizat pentru actualizarea produsului (TECHNOLOGY, 2014, pp. 1-5).

Cum funcționează DevOps?

Ca toate culturile, DevOps încorporează mai multe variante. Cu toate acestea, majoritatea observatorilor ar fi de acord că următoarele capacități sunt comune practic tuturor culturilor DevOps: colaborare, automatizare, integrare continuă, livrare continuă, testare continuă, monitorizare continuă și remediere rapidă.

Colaborare. Operațiile de dezvoltare și IT lucrează împreună. În timp ce deconectarea dintre aceste două grupuri a fost un impuls pentru crearea sa, DevOps se extinde mult dincolo de departamentul IT, deoarece nevoia de colaborare se extinde tuturor celor cu o participare la livrare.

Automatizare. DevOps se bazează foarte mult pe automatizare și instrumente, care se construiesc, care se cumpără sau se obțin din surse open. În plus față de instrumente, DevOps se bazează pe cablurile propriu-zise, care leagă aceste instrumente, pentru a automatiza părți mari din procesul de dezvoltare și implementare a software-ului end-to-end.

Integrare continuă și livrare continuă (CI / CD). Integrarea continuă înseamnă că dezvoltatorii îmbină regulat modificările codului.

Infrastructura cod. În mod tradițional, informațiile despre configurația infrastructurii erau stocate într-unul din două locuri: memoria administratorului de sistem sau a unei mașini (de unde putea fi citită doar de un administrator de sistem). Dar o persoană se poate îmbolnăvi și o mașină poate fi furată sau spartă. De aceea, stocarea informațiilor trebuie făcută astfel, încât să nu depindă de vulnerabilitatea unui expert sau hardware și orice membru al echipei să poată avea acces. .

Infrastructura ca practică de cod implică faptul că resursele sistemului și setările infrastructurii sunt stabilite prin fișiere de definiție prelucrate automat, ceea ce permite dezvoltatorului DevOps să le configureze ca și cod de program. Aceste fișiere pot fi stocate în același sistem de control al versiunii, unde este stocat codul, astfel încât pot fi, de asemenea, revizuite și returnate. Mai mult, modificările fișierelor de configurare sunt implementate automat în mediul de producție (Riungu-Kallosaari, 2016, pp. 25-30).

Microservicii. Arhitectura „microservicii” presupune că aplicația este împărțită în părți mici, fiecare dintre ele construită în jurul unei funcții elementare. Ele pot fi chiar scrise în diferite limbaje de programare, utilizând cadre diferite și funcționează sub diferite sisteme de operare.

Microserviciile sunt mai ușor de testat, întreținut și reutilizat. Dependentele dintre ele sunt reduse la minimum, astfel încât adăugarea unei noi nu necesită nicio schimbare în piesele existente. Cu toate acestea, această abordare nu poate funcționa fără teste concepute cu gândire. Un proiect construit pe microservicii este întotdeauna gata pentru creștere și schimbări semnificative (S.W.Ambler, 2011, pp. 18-23).

Managementul configurației. Această practică necesită înregistrarea și actualizarea detaliată a informațiilor, descrierea hardware-ului și software-ului (actualizări lansate, versiuni, setări de mediu etc.). Mai simplu spus, compania deține întotdeauna informații actualizate pentru toate resursele sale. Aceste înregistrări oferă o asigurare excelentă în cazul în care apar probleme majore, deoarece se pot efectua întotdeauna revizii și se poate recupera orice versiune de configurare. Închiderea („căderea”) serverului chiar și pentru o secundă poate duce la pierderi semnificative de bani pentru companii, de aceea gestionarea configurației trebuie să confere sistemului o fiabilitate ridicată.

Livrare continuă. Echipa de la AWS (Amazon Web Services) definește livrarea continuă ca o practică de dezvoltare software DevOps în care modificările de cod sunt automat construite, testate și pregătite pentru o lansare în producție. Această definiție se extinde la integrarea continuă prin implementarea tuturor modificărilor de cod într-un mediu de testare și/sau un mediu de producție după etapa de construire. Când livrarea continuă este implementată în mod corespunzător, dezvoltatorii vor avea întotdeauna un artefact de pregătire pentru implementare, care a trecut printr-un proces de testare standardizat.

Frecvența de lansare reală poate varia foarte mult în funcție de obiectivele companiei. Companiile performante care utilizează DevOps realizează mai multe implementări pe zi, comparativ cu companiile care au performanță medie, care lansează între o dată pe săptămână și o dată pe lună.

În unele organizații, QA (asigurarea calității) și operațiunile eliberează potențialul de triaj; multe se adresează direct utilizatorilor, unele revin la dezvoltare, iar câteva pur și simplu nu sunt deloc implementate. Alte companii trimit tot ceea ce vine de la dezvoltatori către utilizatori și mizează pe monitorizare în timp real și remediere rapidă pentru a minimiza impactul unui eșec. Este important de reținut că, deoarece fiecare actualizare este mai mică,

șansele ca oricare dintre ele să provoace o defecțiune sunt reduse semnificativ (S.W.Ambler, 2011, pp. 18-23).

Monitorizare și fișierele de log . Niciun test nu poate furniza informații cu același grad de fiabilitate și volum la fel ca datele reale de la utilizatorii reali. Adunarea, clasificarea și analizarea fișierelor de log permite echipei să detecteze probleme, să vină cu noi funcții și să îmbunătățească continuu produsele și procesele.

Valori cum ar fi „timpul mediu pe pagină” sau „vârsta medie a utilizatorului” pot contribui la a înțelege mai multe despre audiența și clienți potențiali, furnizând idei despre ce modificări ar trebui făcute pentru a crește vânzările sau pentru a simplifica UX (user experience design).

Instrumentele DevOps

Pentru fiecare practică există un set de instrumente DevOps care pot fi utilizate. Câteva astfel de exemple sunt prezentate în Figura 4, acestea fiind implementate și folosite de după inginerii DevOps de exemplu în cadrul companiei Raiffeisen Bank.



Figura 4. Instrumente utilizate de către inginerii DevOps (Stars, 2020)

- Docker – folosit pentru „containerizare”. Containerizarea permite dezvoltatorilor să creeze un software independent de mediul în care urmează să ruleze. Software-ul este ulterior ambalat în unități (containere) standardizate care pot fi implementate în orice mediu.

Cu siguranță, Docker nu este singurul instrument pentru asta, dar este cel mai popular.

- Jenkins și Circle CI - pentru integrare continuă și livrare continuă
- Sentry - pentru urmărirea în timp real a erorilor
- Terraform și Ansible - pentru furnizarea de server și infrastructură
- Dynatrace - pentru monitorizare
- Git (GitHub) – pentru gestionarea și versionarea codului
- Jira – pentru gestionarea muncii, de la cerințe și gestionarea cazurilor de testare până la dezvoltarea de software agil
- AWS (Amazon Web Services) – pentru livrarea la cerere a resurselor IT

Certificarea DevOps confirmă cunoașterea unui anumit instrument sau a unui set de instrumente.

DevOps în Facebook

Facebook a lansat inițial funcții precum cronologie, muzică și ticker . Miliarde de oameni erau pe server în ziua implementării, la nivel global, a funcțiilor precizate anterior. Dar, din cauza traficului intens serverul a „căzut”. Atunci a fost lansat Facebook „the dark technique”, care inițial se bazează pe caracteristica DevOps de dezvoltare, testare, monitorizare și implementarea continuă (Baker, 2016).

Extinderea DevOps creează cerere de talent

Datele de literatură confirmă cererea puternică pentru profesioniștii DevOps, deoarece un număr tot mai mare de organizații încep să implementeze modelul de dezvoltare în organizațiile lor, a declarat (Vijayan), un analist al grupului 451 . „Am văzut că DevOps s-a răspândit dincolo de companiile web și întreprinderile de sânge la un public de întreprindere mai mainstream, care include aproape toate verticalele majore." (DevOps Salary Report: IT back in demand as DevOps matures, spreads, 2019)

Profesioniștii IT din mediile DevOps tind să obțină salarii mai mari parțial deoarece, de asemenea, tind să aibă mai multe responsabilități care se referă la dezvoltare și operațiuni IT. DevOps a făcut ca organizațiile să fie mai rapide și mai eficiente pentru a rămâne competitive.

Adoptarea din ce în ce mai mare parte a DevOps duce la intensificarea cererii pentru abilități abia disponibile în prezent, în domenii precum automatizarea și livrarea aplicațiilor cu viteză și agilitate.

Concluzii

Lucrarea de față a descris pe scurt introducerea, evoluția, tipurile și componentele DevOps. Analiza efectuată în lucrare demonstrează câteva avantaje și dificultăți asociate cu DevOps. Avantajele includ descărcări mai continue, mecanizare îmbunătățită a testelor. DevOps este implementat rapid, organizațiile fiind dornice să profite de o livrare mai rapidă a aplicațiilor, o inovație sporită, medii de operare mai stabile și echipe de angajați orientate spre performanță. De aceea este o cerere în creștere pe piața muncii de ingineri DevOps.

Bibliografie

- Baker, J. (2016, April 22). Why Leading Companies Dark Launch. Retrieved from Dzone:
<https://dzone.com/articles/why-leading-companies-dark-launch>
- David, C. (2014). Introduction to DevOps am AWSI in Amazon Web Services. In D. Chapman, Introduction to DevOps am AWSI in Amazon Web Services (pp. 1-20).
- Earnshaw, A. (2013, 5 23). What Is a DevOps Engineer? Retrieved from Puppet:
<https://puppet.com/blog/what-a-devops-engineer/>
- Feijter, R. d. (2016). Towards the adoption of DevOps in software product organization: A maturity model approach. In R. d. Feijter.
- Khaled, A. M. (2016). Tools and Practices to Enhance DevOps Core Values.
- Liu, S. (2019, 08 9). State of DevOps 2019. Retrieved from State of DevOps:
<https://services.google.com/fh/files/misc/state-of-devops-2019.pdf>
- McNeill, S. (2019, Feb 2019). What is DevOps? Why use it? Retrieved from Github:
<https://github.com/letslearncomputerscience/DevOpsAWS/docker/wiki/What-is-DevOps%3F-Why-use-it%3F>
- Moy, E. (2019, 10 18). DevOps Tools of the Trade. Retrieved from Osolabs:
<https://www.osolabs.com/blog/devops-tools-of-the-trade/>
- Nair, A. (2019, October 8). How do I be a good DevOps engineer. Retrieved from Quora:
<https://www.quora.com/How-do-I-be-a-good-DevOps-engineer>
- Riungu-Kallosaari, L. (2016). DevOps BENEFITS AND CHALLENGES IN PRACTICE.
- S.W.Ambler. (2011). disciplined agile delivery and collaborative DevOps in Cutter IT Journal.
- Stars, D. (2020). What is DevOps and Why You Should Have It. Retrieved from djangostars.com: <https://djangostars.com>

TECHNOLOGY, S. (2014, Octombrie). Why DevOps Matters: practical insight on managing complex& continuous change in Microsoft, pp. 1-5.

Vijayan, J. (2019, 11 22). DevOps Salary Report: IT back in demand as DevOps matures, spreads. Retrieved from Techbeacon: <https://techbeacon.com/enterprise-it/devops-salary-report-it-back-demand-devops-matures-spreads>

Studii și Cercetări Academice

Volum 2, Numărul 1, 2020

ISSN: 2668-599X

studii-cercetari-academice.org

Editor
Victoria Folea, Ph.D.
Universitatea Romano-Americana