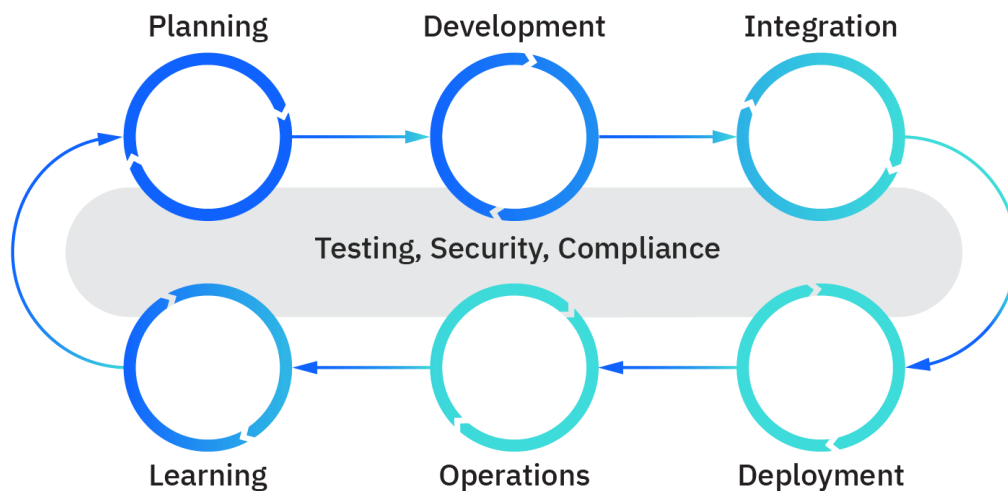


1) Om yrkesrollen DevOps funktion inom systemutvecklingsprojekt och i linje organisationer.

Devops är en uppsättning av metoder som används för att skapa samarbete och klarhet inom mjukvaruutveckling och IT- verksamhet och bildar större synlighet för affärsteam och i slutändan mer intäkter. Den är avsedd att förbättra arbetet under hela livscykeln för mjukvaruutveckling. Det går att tänka sig en DevOps process som en oändlig slinga, som består av dessa steg: planera, koda, bygga, testa, släpp, distribuera, drifta, övervaka och - genom återkoppling - planera, vilket återställer slingan. Förbättrad kommunikation och stramning av återkopplingsslingor mellan IT och utvecklare hjälper till att snabbt bygga observerbara system som driver verkligt kundvärde.



En av de viktigaste roller hos en DevOps-ingenjör är att vara en bra kommunikatör som exempelvis en IT-projektledare då de vanligtvis arbetar i ett DevOps-team med andra proffs i en liknande roll där var och en hanterar sin egen bit av infrastrukturpusslet. DevOps sammanlänkar klyftan mellan mjukvaruutvecklings team och IT-verksamhet. Man kan se på dessa avdelningar som två sidor av samma mynt som ibland har svårt att dela varandras vision.

Linjeorganisation är den enklaste ramen för hela den administrativa organisationen. Det är en typ av organisation med en hierarkisk ordning som fördelar ansvar till separata funktionella enheter, som till exempel marknadsföring och produktutveckling. I en linjeorganisation har varje företag sina egna utmaningar men de vanligaste problem inom företagen brukar bland annat vara utgåvor som tar för lång tid eller mjukvara som inte uppfyller förväntningarna samt IT som begränsar företagets tillväxt.

Utan väntetider, manuella processer och långa granskningar hjälper DevOps ett projekt att lämna alla krav till att distribuera en mjukvara snabbare. Kortare cykeltider kan hålla kraven från att förändras så att produkten levererar det kunderna vill ha (Techtarget u.å).

2) Om DevOps-rollens syfte och mål i projekt och organisationer, inklusive affärs perspektivet / entreprenörskap

DevOps löser kommunikation och prioriterade problem mellan IT -specialiseringar. Det är DevOps -ingenjörans ansvar att inkludera och ändra affärsprocesser efter behov för att lösa organisatoriska problem och förbättra affärsresultat. Med en DevOps-kultur svarar inte utvecklarna att "Det fungerade på min maskin" när ett problem uppstår. Förändringar som rullas ut till produktionen är små och reversibla. Dessutom förstår hela teamet förändringarna, så incidenthantering är mycket förenklad.

DevOps leder till en snabbare process från idé till levande programvara vilket kan resultera i olika marknadsmöjligheter. På detta sätt ger DevOps en konkurrensfördel mellan olika företag.

Det finns fem huvudsakliga anledningar till att man bör involvera en DevOps-ingenjör i sitt företag:

-Förbättrad flexibilitet

Programvaruutvecklings behov förändras snabbt i vår värld. DevOps- ingenjör hjälper till att snabbt svara på förändringar och implementera nya funktioner som hanterar moderna utmaningar.

-Ökad operativ effektivitet

Huvudmålet med DevOps är att minska mjukvaruutvecklings tiden och påskynda lanseringen av uppdateringar som gör operativa processer mer dynamiska och förutsägbara och maximerar deras effektivitet och säkerhet.

-Högre intäktsggrad

Enligt EMA har företag som använder DevOps-strategin 11 gånger större chans att se tvåsiffriga årliga intäktstillväxt.

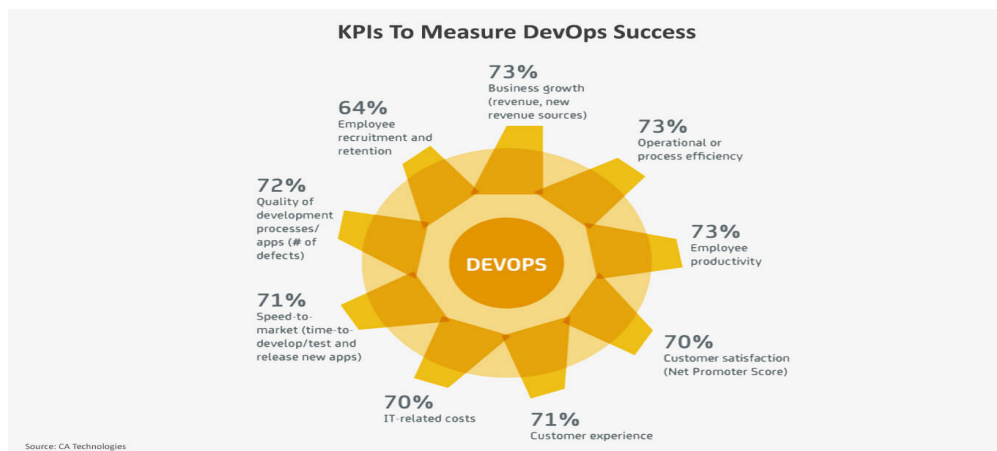
-Förbättrad kundnöjdhet

DevOps kan hjälpa företag att förbättra kundnöjdheten. Förbättrad kundservice resulterar i ökad kundlojalitet, kvarhållande, förbättrad försäljning och ökad avkastning.

-Ökad konkurrenskraft

DevOps ökar mjukvarukvaliteten och minskar släpptiden för uppdateringar som gör din produkt mer konkurrenskraftig. Det låter en också få snabb feedback och göra nödvändiga förbättringar (Exposit, 2021).

Vissa organisationer utvidgar DevOps omfattning till att omfatta andra avdelningar. I DevSecOps sker säkerhetsplanering, skanningar, tester och recensioner kontinuerligt under hela DevOps-slingan. BizDevOps fokuserar på att ansluta chefer, programägare och andra affärsintressenter till det tekniska teamet, som utvecklar, testar och stöder programvaran. Medan mer samarbete utan tvekan alltid är bättre än mindre, måste dessa samarbetspartners dela effektivt och i rätt tid.



Här är en indikation på DevOps fördelar för företag och organisationer (DevOps for Business: Ultimate Guide For 2020).

3) Om de vanligaste rollerna inom IT-branschen och vart roller i olika fall opererar i teknikstackarna.

Ett IT-team är beroende av många olika specialiserade kunskaper inom information och teknik för att stödja utrustning, applikationer och aktiviteter. IT- arbetare kan specialisera sig inom olika områden såsom mjukvaruutveckling, applikationshantering, hårdvarukomponenter, server, lagring eller nätverksadministration, nätverksarkitektur med mera. De flesta företag söker IT-proffs som har mera blandade färdigheter (Castagna & Bigelow).

Det finns ett brett utbud av IT-karriärer men de vanligaste IT rollerna är följande:

- Informationschef (CIO) som är ansvarig för IT- och datorsystem som stödjer verksamhetens mål.
- Teknisk chef (CTO) som anger teknologimålen och policyn inom en organisation.
- IT-chef eller IT- ledare som är ansvarig för funktionen av företagets tekniska verktyg och processer.

- Systemadministratör (sys admin) som konfigurerar, hanterar, stödjer och felsöker en datormiljö för flera användare. Inom ett företag kan denna roll delas upp efter teknik, vilket kräver en administratör eller ett team som är dedikerat till server, skrivbord, nätverksadministration, virtualisering eller andra komponenter och tekniker.

-Application manager. Den här personens roll handlar om tillhandahållande och hantering av en affärsapplikation med högt värde, till exempel Exchange.

- Utvecklare eller mjukvaruingenjör som uppdaterar och testar kod för datorprogram för att uppfylla interna eller kundinriktade affärsmål.

- IT -arkitekt som undersöker och ändrar IT -funktioner för att bäst stödja verksamheten (New Horizonz, 2018).

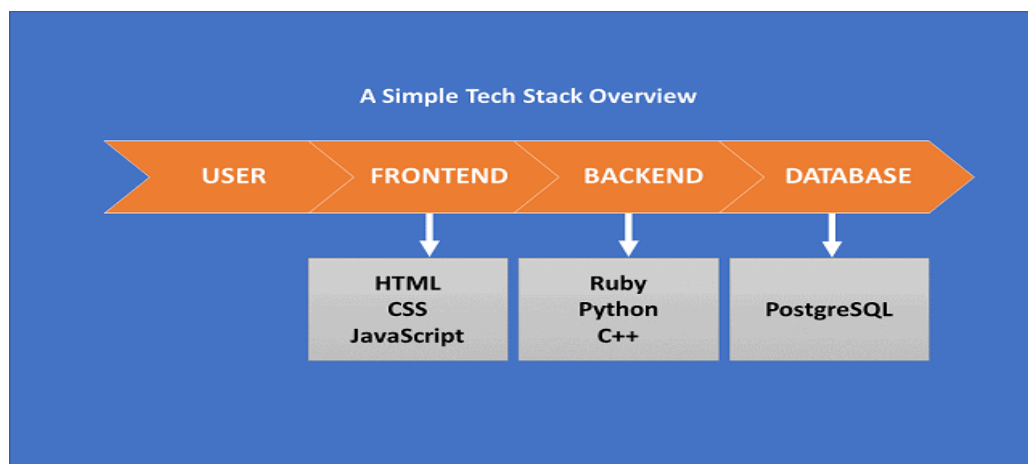
- Webbutvecklare som ansvarar för att skapa webbsidor.

-Dataprogrammerare som skriver koden som tillåter att programmet kan köra.

En fullständig teknikstack innehåller alla aspekter av IT- infrastruktur som krävs för att distribuera och hantera applikationer och tjänster: maskin- och programvarukomponenter, databaser, mellanprogram, lagring och nätverk. I modern IT -företag utgör mjukvara allt fler av dessa stacks. Det finns många typer av teknikstackar. Vissa är konstruerade för att leverera en teknikprodukt eller tjänst, medan andra stöder ett affärsändamål. Exempel inkluderar: Server stack, storage stack, cloud infrastructure stack och marketing stack.

4) Om de mest använda teknikstackarna inom systemutveckling.

En teknikstack är grovt delad i två delar: front-end och serversidan (back-end). Back-end-teknik inkluderar webbapplikationer, programmeringsspråk, servrar och operativsystem och front-end-teknik inkluderar HTML, CSS, JavaScript och UI-framework och bibliotek.



En teknikstack är som blodet i någon programvara. En utvecklare ska veta om teknikstacken i en app innan han gör några förbättringar eller skapar en intelligande produkt för att arbeta med en specifik app. Att välja en relevant teknikstack är ett kritiskt beslut man måste ta innan ett program startas. Den kan exempelvis påverka hur ens programvara fungerar eller kommer att fungera i framtiden, programvarans skalbarhet, målet med ens företag/projekt, Valet av app- och datalagring (lokalt eller moln) samt möjligheten för ens serversystem (Pratik Mistery, 2021).

De mest använda teknikstackarna inom systemutveckling är:

1. The *MEAN* stack

MEAN är en gratis JavaScript-programvarustack med öppen källkod för att bygga dynamiska webbplatser och webbapplikationer. MEAN är en av de populära teknikstackarna som har en mängd fördelar för en webbutvecklare, exempelvis att man kan använda sig av ett enda språk i hela stacken. Komponenterna i denna teknikstack är: MongoDB (NoSQL -databas), Express.js (backend -ramverk), Angular.js (frontend -ram), NodeJS (öppen källkod, plattformsserver). Med Mean stack kan man bygga snabba, flexibla och utökbara programvaruprodukter då den är perfekt för molnhotell.

2. The *LAMP* stack

LAMP stack är en av de vanligaste lösningstackarna för många av webbens mest kända applikationer. I webbprogramvaruutveckling ger LAMP den bästa prestanda, stabilitet, flexibilitet och kostnadseffektivitet. LAMP-komponenter är: Linux (operativsystem), Apache (HTTP -server), MySQL (relationell databashantering), PHP (programmeringsspråk, det kan också vara Perl eller Python). LAMP stack kan köras på alla operativsystem, vilket ger en chansen att ändra eller byta ut andra komponenter i samma stack. I stället för att använda Linux som operativsystem har man möjligheten att använda MAMP för MacOS och WAMP för Microsoft Windows.

3. The *MERN* stack

MERN är en Javascript stack som används för en enklare och snabbare distribution av full-stack webbapplikationer. Det är en av flera varianter av MEAN stacken och är i grunden mycket lik MEAN då den enda skillnaden är att Angular JS ersätts av React JS i MERN. Komponenterna i MERN är: MongoDB (databas), Express JS, React JS och Node JS. MERN tillhandahåller hela front-end till back-end utvecklingskomponenter då det är ett open source-utvecklingsramverk för fullstack.

Med MERN kan du enkelt använda React-integration, samtidigt använda kod på servrar och webbläsare, få tillgång till utvecklingsalternativ med fullstack (backend och frontend) och tillgång till kraftfullt bibliotek.

4. The MEVN stack

MEVN -stack är en open source för en Javascript programvara och en annan populär version av MEAN som använder Vue.js som frontend-framework istället för Angular.js. Vue.js har fått mycket erkännande under de senaste åren då den erbjuder abstrakta lösningar. MEVN komponenter är: MongoDB (NoSQL -databas), Express.js (backend -ramverk), Vue.js (frontend -ramverk), NodeJS (öppen källkod, plattformsserver)

5. Ruby On Rails

RoR är en integrerad utvecklingsmiljö som använder Ruby, ett kraftfullt programmeringsspråk i 2021. Denna ram för webbapplikationer kan förenkla hela utvecklingsupplevelsen. RoR är mycket skalbar och gör det möjligt för utvecklare att använda och ändra det utan några begränsningar (Arun Thomas, 2021)

5. Om hur ett modernt IT-system är uppbyggt inom de större teknikstackarna och större skillnader av användningsområden för olika teknik.

Att välja rätt teknikstack är mycket kritiskt och kan påverka utvecklingstid, kostnad, applikationskvalitet och skalbarhet. För att fatta ett rätt beslut måste man lägga mer tid på att analysera fördelar och nackdelar med tillgängliga lösningar. Först och främst, innan man väljer en teknikstack, är det viktigt att ta hänsyn till projektets storlek och syfte. Större och mer komplex projektet innebär även mer komplex teknikstack.

För små projekt kan sådana väletablerade staplar som Python-Django eller Node.js-React vara perfekta. För Medelstora projekt som onlinebutiker eller krävs det vanligtvis mer komplexa tekniska staplar med flera lager programmeringsspråk och ramar och stora projekt, som komplexa marknadsplatser eller företagsapplikationer, kräver en stor och mycket mer komplex teknikstack.

IT -avdelningen kontrollerar och ser till att organisationens system, nätverk, data och applikationer och alla komponenter i techstacks fungerar bra. IT -teamet hanterar tre huvudområden: De distribuerar och underhåller affärsapplikationer, tjänster och infrastruktur (servrar, nätverk, lagring); övervakar, optimerar och felsöker prestanda för applikationer, tjänster och infrastruktur samt övervakar säkerheten och styrningen av applikationer, tjänster och infrastruktur (Unit Space).

6. Om CD/CI samt dess anknytning till modern agil systemutveckling och DevOps roll i det avseendet.

Agile mjukvaruutveckling är en metod för systemutveckling som fokuserar på leverans av enskilda delar av programvaran, alltså inte på hela applikationen medan CD/CI handlar mer om metoder och verktyg (Kate Brush).

CD (Continuous delivery) är en metod som ser till att testa och verifiera koder i korta men kontinuerliga cykler på ett automatiserat sätt och ger direkt feedback om koden fungerar som den ska (Gillis. A).

CI (Continuous integration) är en utvecklingsmetod där frekventa förändringar testas omedelbart och rapporteras när de läggs till i en större kodbas (Gillis. A)

DevOps är en uppsättning arbetsprocesser, metoder och verktyg som gör processen från utveckling till drift och underhåll av programvara effektivare.

Den kräver ett nära samarbete mellan verksamheter och olika utvecklingsteam i varje fas av service livscykeln från design till utveckling och produktion. Devops uppstod från Agiles framgång med att förbättra utvecklingshastigheten, när de blev klart att kopplingar mellan utvecklings- och driftsteam liksom mellan IT och affärssidan i organisationen hindrade Agile mjukvarans leverans till användarna (Kelly, 2019)

7. Om hur Python används inom agil systemutveckling och automatiserade flöden.

Vad är agil utveckling?

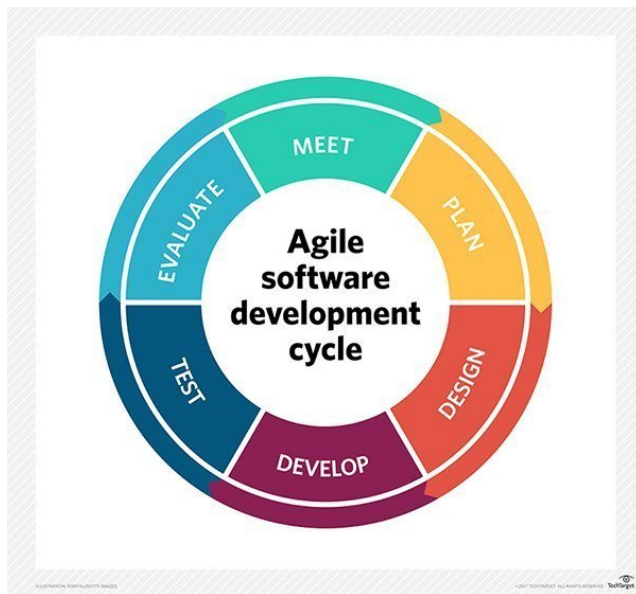
Agil utveckling är en term som ges till en hel klass av iterativa utvecklingsmetoder. Deras förenande egenskap är fokus på *korta utvecklingscykler*, i skala av veckor snarare än månader. Varje utvecklingscykel, kallad en iteration eller sprint, producerar en fungerande produkt.

Vad är agila metoder?

Agila metoder är en samling av olika tekniker som kan användas tillsammans för att uppnå hög mjukvarukvalitet och exakta uppskattningar av tid och material med *korta utvecklingscykler*. De agila metoderna fokuserar på olika delar av livscykeln för utvecklingen av mjukvaran. Vissa metoder fokuserar på metoder som extrem programmering och pragmatiskt programmering, medan andra metoder handlar om att hantera mjukvaruprojekt. Dock finns det även metoder som täcker upp en mjukvaru hela livscykel.

När jag tänker på Agile tänker jag på detta ordspråk: "Vi kan inte ändra vindens riktning. Vi kan bara justera våra segel." Och det seglet kan vara Python som på grund av sina fördelar hjälper till att förkorta utvecklingscykeln, eftersom Python är ett programmeringsspråk på hög nivå som har engelska-liknande syntax, detta gör det lättare att läsa och förstå koden och Python är verkligen lätt att plocka upp och lära sig, Python är ett mycket produktivt språk. På grund av Pythons enkelhet kan utvecklare fokusera på att lösa problemet. De behöver inte lägga alltför mycket tid på att förstå syntaxen eller beteendet hos programmeringsspråket. Du skriver mindre kod och får fler saker gjorda. Python är kraftfullt, flexibelt och bland de mest populära och snabbast växande språken i världen. Det används i många organisationer

eftersom det stöder flera programmeringsparadigm. Det utför också automatisk minneshantering (Naor, 2020)



8. Om hur Python kan hjälpa inom skriptprogrammering, automatiserade DevOpsflöden, systemautomation, automatiserade testning och webbservernskriptning.

Python är ett programmeringsspråk men den kan också användas för skript. Skriptspråk skriver en Python kod som är enkelt att förstå även för nybörjarna inom IT. Python används mycket i programmering och har ett tolkande språk som förekommer under körning. Då Python översätter kod är den mer flexibel än många andra skriptspråk. Python anses vara ett skriptspråk då den översätter källkoder.

Automation är en process där en manuellt utförd åtgärd omvandlas till en som sker automatiskt vilket leder till att en uppgift kan köras mycket snabbare och att risken för mänskliga misstag minskas.

Python är ett mångsidigt programmeringsspråk som är relativt lätt att lära sig och använda. Den har en lättillgänglig syntax som är lätt att förstå i förhållande till andra programmeringsspråk. Eftersom Python är en öppen källkod har man tillgång till stort utbud av verktyg, bibliotek, ramverk och stöd för det. Därför kan Python vara ett bra val av programmeringsspråk om man önskar köra enkla automatiseringar. Det enda man ska göra är att ta reda på vilka verktyg man behöver och hur man kan skriva och köra en skript.

Det finns ett stort utbud av uppgifter som man kan välja att automatisera genom att skriva ett Python-skript. Några av de vanligaste vägar som Python automatisering används är att skicka ut, svara på och sortera mail, fylla i PDF-filer och Excel-filer, skicka förfrågningar, konvertera bildfiler, utföra snabba matematiska ekvationer med mera. Några verktyg som kan användas för att göra automatiseringar enklare och mer intuitiva är exempelvis:

-Smtplib - biblioteket som hjälper med att automatisera e-postmeddelanden genom att använda simple mail transfer protocol, som kan integreras med de flesta större e-postplattformar, som Gmail.

-Selenium är ett programvara med öppen källkod som man kan använda för att automatisera tester som utförs i webbläsare.

-Beautiful Soup är ett Python - bibliotek som kan användas för att dra ut data från HTML- och XML -filer.

Pytest är ett testramverk som tillåter användare att skriva testkoder med Python programmeringsspråk samt är en av de bästa tillgängliga ramerna för automatisering som finns. Pytest hjälper en att skriva enkla testfall för databaser och kan hantera alla funktionstest, oavsett om vi pratar om enhet, integration eller end-to-end-testning. Testfall skrivs helt enkelt som funktioner och kan ta parametriserade ingångar. Plugins utökar pytest-funktioner och låter dig täcka kod, köra flera test samtidigt och integrera med andra ramverk, som Django och Flask(Abi Nakhoul & how to Python)

Referenser:

Meredith. C & Alexander S.G. What is DevOps? The ultimate guide från,
<https://searchitoperations.techtarget.com/definition/DevOps>

Exposit. (9 februari 2021). What is the role of a DevOps Engineer in a software development project?
<https://www.exposit.com/blog/devops-engineer-in-software-development-project/>

Softermii. (3 January 2020) . DevOps for Business: Ultimate Guide For 2020.
<https://www.softermii.com/blog/devops-for-business-ultimate-guide-for-2020>

New Horizonz. (9 februari 2018). 7 IT Roles Every Modern Company Needs to Stay Competitive.
<https://www.newhorizons.com/article/7-it-roles-every-modern-company-needs-to-stay-competitive>

Castagna. R & Bigelow. S. Information technology (IT)
<https://searchdatacenter.techtarget.com/definition/IT>

Mistry.P. (16 Juni 2021) Top 5 Technology Stacks To Nail Your 2021 Software Project.
<https://pratikmistry.medium.com/top-5-technology-stacks-to-nail-your-2021-software-project-ca26d8ab0ad8>

Arun. T (16 Mars 2020). Top 6 Tech Stacks That Reign Software Development in 2021.
<https://www.fingent.com/blog/top-6-tech-stacks-that-reign-software-development-in-2021/>

Unit Space (19 Juli 2021). What is Tech Stack and how to choose the best one for your project.
<https://medium.com/unit-space/what-is-tech-stack-and-how-to-choose-the-best-one-for-your-project-10decc674f72>

Brush, K. & Silverthorne. V. Agile Software Development
<https://searchsoftwarequality.techtarget.com/definition/agile-software-development>

Gillis. A.S. continuous delivery (CD)
<https://searchitoperations.techtarget.com/definition/continuous-delivery-CD>

Gillis.A.S. Continuous integration (CI)

https://searchsoftwarequality.techtarget.com/definition/continuous-integration?_gl=1*17cwlzd*_ga*MTkxNjkzNzg5NS4xNiMyMjQ3ODMx*_ga_TQKE4GS5P9*MTYzNDE1MjU3MC4zNy4xLjE2MzQxNTU2OTQuMA..&_ga=2.62043244.199745855.1633988295-1916937895.1632247831

Kelly. W. (6 november 2019). Demystify the DevOps process, step by step <https://searchitoperations.techtarget.com/tip/Demystify-the-DevOps-process-step-by-step0>

Naor. A. (8 september 2020). Agile Methods and Methodology for Beginners – Agile Software Development and Agile Project Management Example.
<https://www.freecodecamp.org/news/agile-methods-and-methodology-for-beginners/>

How to Python. (20 April, 2020). Is Python a Programming Language or Scripting Language?
<https://www.howtopython.org/is-python-a-programming-language-or-scripting-language/>

Abi Nakhoul. N . How is Python used in automation?
<https://www.lighthouse labs.ca/en/blog/how-python-is-used-in-automation>