

# DOE21\_ITTO

## U01-ITTO-REDOGÖRELSE

---

### *Innehållsförteckning*

<b>1. Yrkesrollen DevOps funktion inom systemutvecklingsprojekt och i linjeorganisationer.</b>	<b>2</b>
<b>2. DevOps-rollens syfte och mål i projekt och organisationer, inklusive affärsperspektivet / entreprenörskap</b>	<b>4</b>
<b>3. De vanligaste rollerna inom IT-branschen och var dess roller opererar i teknikstackarna.</b>	<b>5</b>
<b>4. De vanligaste teknikstackarna inom systemutveckling</b>	<b>6</b>
<b>5. Hur ett modernt IT-system är uppbyggt inom de större teknikstackarna och större skillnader av användningsområden för olika teknikval.</b>	<b>8</b>
<b>6. Vad är en Agil systemutveckling samt vilken anknytning har CD/CI till denna. Vad är DevOps roll i Agil systemutveckling.</b>	<b>9</b>
<b>7. Python´s användning inom agil systemutveckling och automatiserade flöden.</b>	<b>11</b>
<b>8. Hur kan Python hjälpa inom scriptprogrammering, automatiserade DevOpsflöden, systemautomation, automatiserade testning och webbserverscriptning.</b>	<b>12</b>

### **Syfte:**

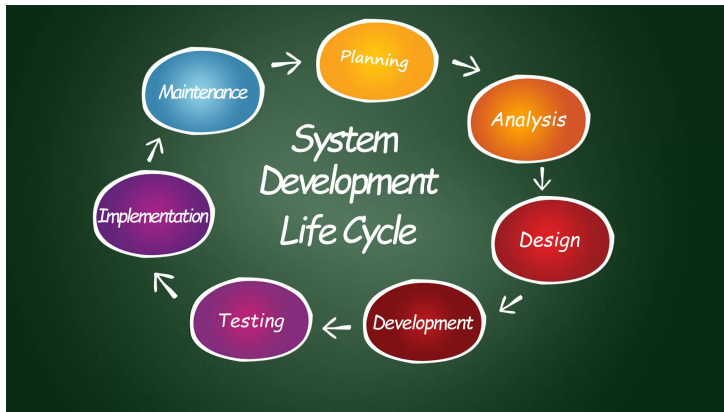
Syftet med denna uppgift är att bekanta dig med hur utveckling och drift av IT-system går till, samt angränsande roller, ur ett teoretiskt perspektiv. Du tränas även i att skriva redogörelser för att vi ska kunna examinera dig i teoretiska kursmål.

### **Sammanfattning:**

För att lösa uppgiften har jag delat upp frågeställningarna i mindre delar som jag sedan har sökt information om på webben. Upplever att det finns mycket material skrivet om ämnet, dock kan svaren i vissa frågor, beroende på vilken källa man läser, skilja sig. En annan notering är att flertalet frågeställningar berör samma "sak" vilket har gjort det svårare att få en "röd tråd" genom redogörelsen.

## 1. Yrkesrollen DevOps funktion inom systemutvecklingsprojekt och i linjeorganisationer.

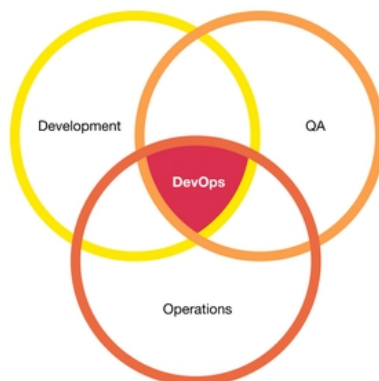
Systemutveckling används på teknikdrivna företag. Syftet med systemutveckling är att öka produktiviteten i organisationen och i den grupp som arbetar i den organisationen. Allt eftersom systemutvecklingen växte blev det nödvändigt att systematisera processen. Resultatet blev en utvecklad steg-för-steg-process som är tydligt beskriven, se nedan bild.



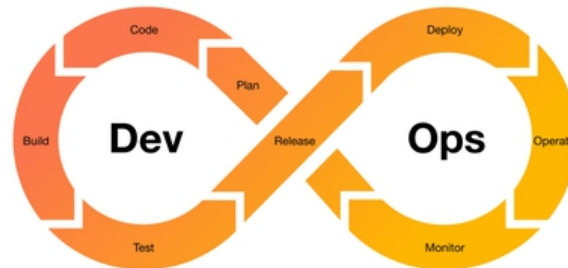
Med linjeorganisation menas en hierarkiskt uppbyggd organisationsform där varje medarbetare endast har en enda direkt överordnad chef. Detta faktum anses vara en fördel med en linjeorganisation. Organisationsformen utmärks av en strikt fördelning av arbetsuppgifter, ansvar och befogenheter. En nackdel anses vara att organisationsstrukturen leder till att beslut tar lång tid dvs att det är en långsam organisation.

### Dev.ops roll i ett systemutvecklingsprojekt och i linjeorganisationer

En DevOps Engineers kombinerar samarbetet mellan de olika kompetensavdelningar, utveckling(development), drift (operation) och kvalitetssäkring(QA) Tack vare DevOps förståelse både för teknik och affärssidan kan det byggas en leveransprocess som är både effektiv samt att den förhindrar oplanerade ändringar.



Man kan likna samarbetet som en DevOps har med utvecklare och testare som en oändlig livscykel där alla delar t.ex planering, kodning, testning, körning och övervakning ingår i livscykeln. Se utförlig bild nedan.



Fördelarna med att engagera en DevOps i mjukvaruutvecklingsprocessen är flera. Till exempel kan nämnas:

- ❖ Förbättrad flexibilitet
- ❖ Ökad operativ effektivitet
- ❖ Förbättrad kundnöjdhet
- ❖ Ökad konkurrenskraft

**Källor:**

<https://www.exposit.com/blog/devops-engineer-in-software-development-project/>  
[http://www.umsl.edu/~sauterv/analysis/488\\_f02\\_papers/methodologies.html](http://www.umsl.edu/~sauterv/analysis/488_f02_papers/methodologies.html)  
<https://smallbusiness.chron.com/examples-system-development-company-79419.html>  
<https://svitla.com/blog/system-development-life-cycle>  
[https://www.bjornlunden.se/f%C3%B6retagande/linjeorganisation\\_\\_1560](https://www.bjornlunden.se/f%C3%B6retagande/linjeorganisation__1560)  
<https://quizlet.com/237921559/organisation-och-organisationsstruktur-iva-flash-cards/>

## *2. DevOps-rollens syfte och mål i projekt och organisationer, inklusive affärsperspektivet / entreprenörskap*

DevOps syfte är att hjälpa organisationer att accelerera sin process fram till en färdig distribution av en produkt.

Traditionellt har processen vid programvaruutveckling sett ut enligt följande:

1. Ett utvecklingsteam bygger kod och testar sedan koden i en isolerad miljö för att kvalitetssäkra produkten.
2. När produkten är färdigutvecklad av utvecklingsteamet släpps den vidare till ett nytt team som är kopplad till produktionen.

I och med att det är två olika team som arbetar helt separerat från varandra leder det till att behovskravet som produktionen har för den färdiga applikationen inte har kommunicerats till utvecklingsteamet. Dessa otydligheter leder till lång leveranstid av färdig-och körbar applikation.

Jämför man det med DevOps arbetssätt ser det ut enligt följande:

1. Utvecklingsteam och produktionsteam jobbar tillsammans i samma team vilket leder till att ansvarsområden som tidigare legat i steg 2 ovan byggs in i utvecklingen tidigare i processen.
2. Bryggan mellan utvecklingsteam och produktionsteam hålls samman genom användande av Dev.Ops verktyg och metoder. Dessa leder till att steg som tex versionskontroll och kontinuerlig prestandakontroll automatiseras. Att teamen arbetar tillsammans gör att organisationer kan släppa en ny applikation inom dagar jämfört med vid tidigare arbetssätt månader.

Genom att använda DevOps arbetssätt förbättrar företag sin IT-prestanda vilket i sin tur leder till att hela organisationen får ett prestandalyft. Ett företag som har implementerat DevOps-metoder kan i och med detta få en konkurrensfördel gentemot andra företag som inte anammat detta arbetssätt.

### **Källa:**

<https://www.digitaldoughnut.com/articles/2016/june/what-is-devops-and-how-does-it-help-organizations>

### *3. De vanligaste rollerna inom IT-branschen och var dess roller opererar i teknikstackarna.*

Det finns många roller inom informationsteknik, IT och olika källor benämner dem på lite olika sätt. Nedan kan ses några exempel:

- Help Desk Support & Help Desk Analyst
- IT -projektledare
- Hårdvarutekniker
- Nätverksingenjör och nätverksadministratör
- Affärsanalytiker
- Systemteknikchef
- IT -chef, CIO/CTO eller vice ordförande

Tittar man på vad källor som Saco listar som inte nämns ovan så kan nedan exempel läggas till:

- Datasäkerhetsansvarig
- It-arkitekt
- Programmerare
- Utvecklare. (Appar, dataspel, m m)
- Webbmaster (ansvarar för att servrar fungerar, sköter om externa webbplatser och interna så kallade intranät)

För att svara på frågan hur ovan roller opererar i teknikstackarna behöver det först förklaras vad definitionen för en "teknikstack" är. Det engelska ordet för en "teknikstack" är "technology stack" och som ordet antyder så är det kopplat till någon typ av teknologi. En definition på vad en "technology stack" är beskrivs, enligt en källa på följande sätt:

- En samling olika teknologier som tillsammans används för att bygga en webb-eller mobilapplikation genom att kombinera tex. programmeringsspråk, bibliotek, programvara och verktyg.

En av den mest kända exemplet på en teknikstack är **LAMP** som vanligtvis används för att bygga en fungerande webbsida. LAMP står för:

- **L**inux – Server operating system
- **A**pache – Web server software
- **M**ySQL – Database system
- **P**HP – Programming language, well-suited for developing interactive web pages

Tillsammans ger ovan teknologier en bra förutsättning för att bygga en fungerande, interaktiv webbsida.

Har man en utvecklarell på ett företag, inom IT, är en möjlighet att man arbetar enligt teknikstacken LAMP. Arbetar man inom datasäkerhet används termen "säkerhetsstackar" som hjälper till att visualisera alla olika verktyg som används för att förhindra cyberhot. Ett exempel på en specifik säkerhetsstack är en "Network Security Stack" som ansluts på nätverket med målet att bland annat analysera nätverkstrafik för att upptäcka och blockera eventuella hot.

#### Källor:

<https://www.netscout.com/what-is/Network-Security-Stack>

<https://www.andplus.com/blog/what-is-a-tech-stack>

<https://www.scoutnetworkblog.com/7-common-types-of-it-roles-and-their-responsibilities/>

<https://stackshare.io/stacks/trending>

<https://www.saco.se/studieval/yrken-a-o/it-yrken/>

## 4. De vanligaste teknikstackarna inom systemutveckling

### The MEAN Stack

MEAN står för

- **M**ongoDB
- **E**xpress.js
- **A**ngularJS
- **N**ode.js

MEAN stacken tillåter utvecklaren att använda samma "språk" både för "frontend- och "backend" utvecklingen. Denna egenskap har gjort MEAN stacken populär.

### The LAMP Stack

LAMP står för:

- **L**inux – Server operating system
- **A**pache – Web server software
- **M**ySQL – Database system
- **P**HP – Programming language, well-suited for developing interactive web pages

LAMP används vanligtvis för att bygga en fungerande webbsida.

### The MERN Stack

MERN står för:

- **M**ongoDB - document database
- **E**xpress(.js) - Node.js web framework
- **R**ead(.js) - a client-side JavaScript framework
- **N**ode(.js) - the premier JavaScript web server

MERN är en av många varianter av MEAN. I MERN har man bytt ut Angular.js mot React.js.

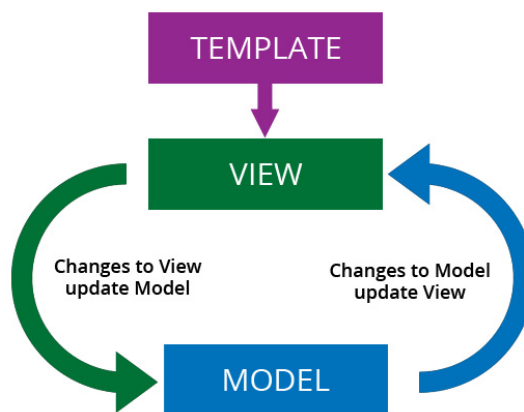
MERN-stacken används med fördel när man som webbutvecklare behöver snabba flöden. Vid tunga arbeten i JSON (JavaScript Object Notation) samt molnlösningar är MERN att föredra. Har man som utvecklare även tidigare erfarenhet av React.js så är valet enkelt.

## The MEVN Stack

MEVN-stacken innehåller:

- [MongoDB](#)
- [Express.js](#)
- [VueJS](#). Används särskilt vid front-end Webbutveckling. Den möjliggör sömlös utveckling genom tvåvägsbindning.
- [Node.js](#)

De it relaterade förändringar som sker i VueJS är tvåvägsbindande. Detta betyder att förändringar som görs, oavsett om det är ändringar gjorda av tex användare så återspeglas till underliggande modellen. Samma sak gäller tvärtom, om appdata ändras så ändras tex gränssnittet.



### Källor:

<https://pratikmistry.medium.com/top-5-technology-stacks-to-nail-your-2021-software-project-ca26d8ab0ad8>

<https://www.performatix.com/what-makes-mean-stack-popular-among-startups/>

<https://www.mongodb.com/mern-stack>

<https://www.educative.io/edpresso/what-is-mevn-stack>

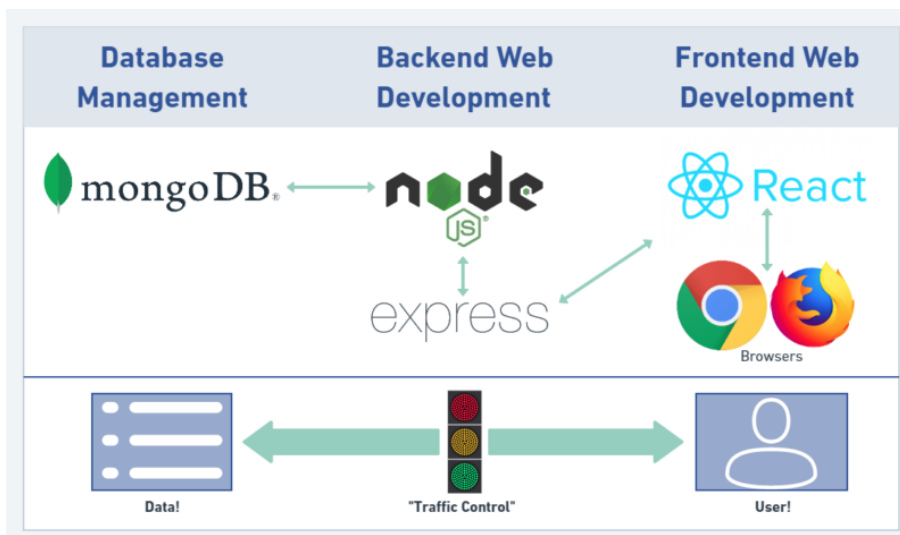
<https://stackoverflow.com/questions/13504906/what-is-two-way-binding>

## 5. Hur ett modernt IT-system är uppbyggt inom de större teknikstackarna och större skillnader av användningsområden för olika teknikval.

Väljer att jämföra stackarna MEAN, MERN och MEVN. Alla tre teknikstackarna används vid utveckling av webbsidor eller mobila applikationer. Nedan ses en överskådlig tabell över likheter och skillnader dessa emellan. Skillnaderna mellan de olika teknikstackarna ligger i vilket front-end-ramverk som dom använder. Exempel på skillnader kan också ses i tabellen.

Stack	MEAN	MERN	MEVN
Databas	MongoDB	MongoDB	MongoDB
Backend	Express.js	Express.js	Express.js
Frontend	AngularJS	React(.js)	<u>VueJS</u>
Server-side programming	Node.js	Node.js	Node.js
Skillnader mellan val av front end-ramverk	Populär inom traditionella organisationer tex. sjukhus och banker. Inte lika anpassningsbart som React och Vue.	Anpassningsbar och uppskalningsbar till att fungera för stora mängder användare. Säkrar framtida tillväxten.	Ett lättläst JavaScript-ramverk. Det är bl.a snabbt och mångsidigt. Vue är relativt nytt och tros vara på framväxt.

En visualiserad bild över MERN-stacken.



### Källor:

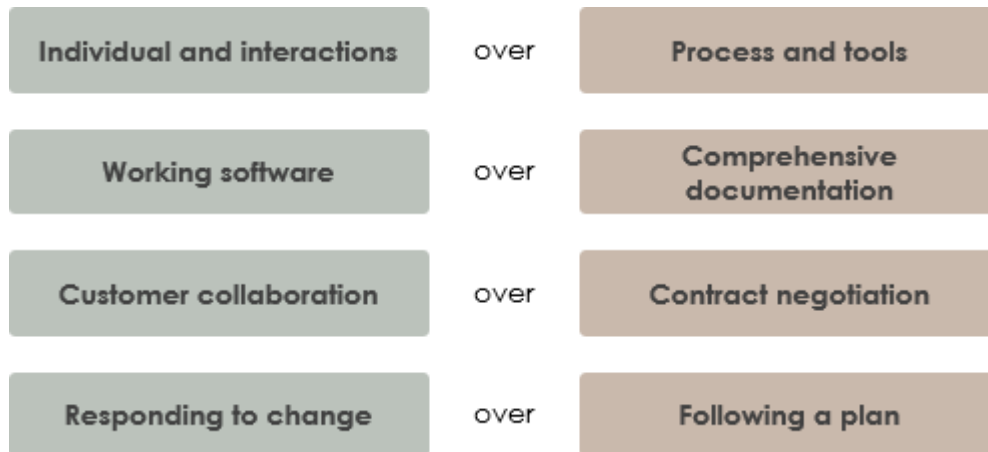
<https://www.kenzie.academy/blog/mean-vs-mern-vs-mevn-stacks-whats-the-difference/>  
<https://www.mongodb.com/mean-stack>  
<https://www.netguru.com/glossary/node-js>



## 6. Vad är en Agil systemutveckling samt vilken anknytning har CD/CI till denna. Vad är DevOps roll i Agil systemutveckling.

### Agil systemutveckling:

Termen "Agile" uppkom 2001 i "Agile Manifesto". Målet var att förbättra tillvägagångssättet vid mjukvaruutveckling genom att bestämma några övergripande principer. Resultatet blev en "Manifesto" bestående av fyra värderingar.



Manifestet har senare kompletterats med 12 underliggande principer. Nedan kan ses dom första 5. Resterande kan läsas i följande länk:

<https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>

1. Vår högsta prioritet är att tillfredsställa kunden genom tidig och kontinuerlig leverans av värdefull programvara.
2. Välkommen att ändra kraven, även sent i utvecklingen. Agila processer utnyttjar förändring för kundens konkurrensfördel.
3. Leverera fungerande programvara ofta, från ett par veckor till ett par månader, med en preferens framför den kortare tidsperioden.
4. Affärsmän och utvecklare måste arbeta tillsammans dagligen under hela projektet.
5. Bygg projekt kring motiverade individer. Ge dem den miljö och det stöd de behöver, och lita på att de får jobbet gjort.

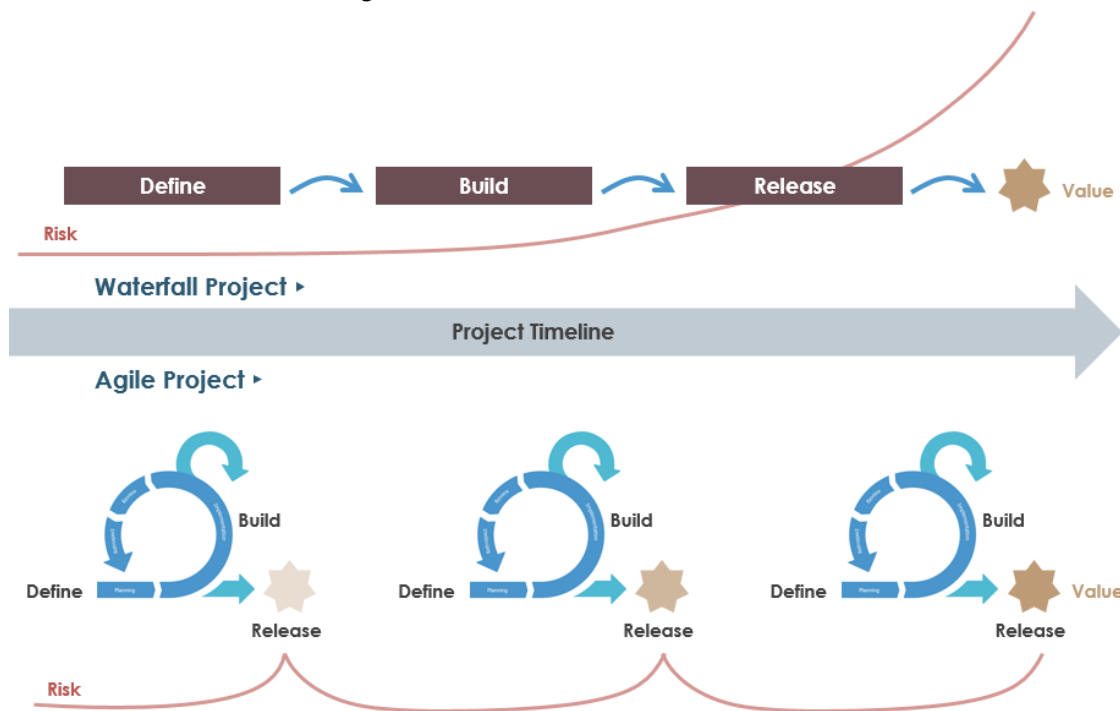
### Waterfall-metoden kontra Agil metodik vid systemutveckling.

Waterfall-metoden är en sekvensuppbyggd, linjär (enkelriktad) modell som består av följande steg:

1. Kravanalys
2. Design
3. Utveckling
4. Implementering
5. Testning
6. Underhåll

Nackdelar med Waterfall-metoden är att man är låst vid varje steg dvs nästa steg kan inte startas förrän steget innan är helt färdigt, det går heller inte att gå tillbaka till steget innan. Varje steg är väldigt tidskrävande och slutprodukten levereras först i sista steget.

Agil är ett annat sätt att tänka och agera efter där man arbetar projektform, i team med blandad kompetens. Ofta arbetar man tvärfunktionellt mellan olika organisationsgrupper för att få bästa möjliga kompetens. Tillvägagångssättet vid Agil systemutveckling bygger på stegvis leverans där man gör små justeringar under processens gång. I stället för att leverera allt på en gång vid slutet av processen så samlas feedback in kontinuerligt genom användning av korta cykler och små "releaser". Nedan är en bild över flödet vid Waterfall-metoden kontra Agil metodik.



### CD(Continuous Delivery)

Continuous Delivery är ett arbetssätt vid leverans av en mjukvara. Koder utvecklas och testas i korta cykler, vanligtvis genom automatisering. Detta sätt att arbeta möjliggör snabba flöden av att bygga kod, testa och släppa program och samtidigt leder det till förbättrad mjukvarukvalitet.

### CD(Continuous Deployment)

Vid Continuous Deployment släpps varje förändring direkt, och automatiskt, till produktionen. Genom detta sätt används användaren som en "live"-testare vilket leder till snabba "releaser".

### CI(Continuous Integration)

Continuous Integration kopplar samman kod från alla utvecklare till en kodbas varefter förändringar testas omgående. CI rapporterar och återkopplar snabbt så att en eventuellt defekt kod identifieras och korrigeras.

## **DevOps roll i Agil Systemutveckling**

Nedan beskrivning finns att hitta under fråga 1. För utökad info gå tillbaka till fråga 1 och läs under **“Dev.ops roll i ett systemutvecklingsprojekt och i linjeorganisationer”**.

“En DevOps Engineers kombinerar samarbetet mellan de olika kompetensavdelningar, utveckling(development), drift (operation) och kvalitetssäkring(QA) Tack vare DevOps förståelse både för teknik och affärssidan kan det byggas en leveransprocess som är både effektiv samt att den förhindrar oplanerade ändringar.”

### **Källor:**

<https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>

<https://it-ord.idg.se/ord/vattenfallsmetoden/>

<https://it-ord.idg.se/ord/agil-systemutveckling/>

<https://searchitoperations.techtarget.com/definition/continuous-delivery-CD>

<https://stackify.com/continuous-delivery-vs-continuous-deployment-vs-continuous-integration/>

## ***7. Python´s användning inom agil systemutveckling och automatiserade flöden.***

### **Python inom Agil systemutveckling**

Vad Agil systemutveckling innebär finns beskrivet under fråga 6, se ovan.

Python är ett bra verktyg att använda för det agila arbetssättet. Vid agil utveckling behöver man använda ett programmeringsspråk som är lätthanterligt för olika typer av personalkompetenser då man i det Agila flödet kopplar ihop tvärfunktionella team. Det behöver också vara kompatibelt med andra program samt ha lätt för att klara många justeringar.

### **Python´s användning vid automatiserade flöden.**

Ett automatiskt flöde beskrivs som en process där en manuell uppgift omvandlas till att ske automatiskt. Fördelarna med en automatisering är att uppgiften kan köras snabbare, minska arbetsbelastningen samt att riskerna för misstag minskas. Python är en mångsidigt programmeringsspråk och anses vara lätt att lära sig. Det faktum att Python har en öppen källkodslicens leder till att det finns ett stort utbud av verktyg och bibliotek. Exempel på olika uppgifter som Python kan automatisera är många. Ett exempel är hanteringen av mail, skicka mail, svara på mail eller sortera mail. Andra exempel är beräkning av växelkurser och konvertera bildfiler.

### **Källa:**

<https://www.lighthouselabs.ca/en/blog/how-python-is-used-in-automation>

## 8. Hur kan Python hjälpa inom scriptprogrammering, automatiserade DevOpsflöden, systemautomation, automatiserade testning och webbserverscriptning.

Ovan frågeställning delas upp i fem olika delfrågor enligt nedan.

Pythons fördelar inom:

- 8a. Scriptprogrammering
- 8b. Automatiserade DevOps-flöden
- 8c. Systemautomation
- 8d. Automatiserande testning
- 8e. Webbserverscript

### 8a. Scriptprogrammering

Det råder olika skolor om huruvida Python är ett så kallat scriptspråk eller ett programmeringsspråk. Enligt källan "How to python" är Python framför allt ett programmeringsspråk, men det kan också användas som ett skriptspråk.

Nedan beskrivs skillnaden mellan scriptspråk och programmeringsspråk samt ger exempel på vilka "språk" som tillhör till varje.

#### Script språk:

- Skriptspråk behöver inte kompileras utan de använder en tolk för att konvertera sin kod till inbyggd maskinkod.
- Skriptspråk går långsammare än programmeringsspråk.
- Skriptspråk kan inte komma åt hårdvaruinformation på låg nivå.
- exempel: JavaScript, Lua, Perl, Python, PHP, VBScript etc.

#### Programmeringsspråk:

- Behöver en kompilator för att konvertera sin kod till inbyggd maskinkod.
- Är snabbare jämfört med skriptspråk
- Dessutom kan programmeringsspråk komma åt hårdvaruinformation på låg nivå.
- exempel på programmeringsspråk. Java, Kotlin, CPython, C, C++, Scala, Groovy etc.

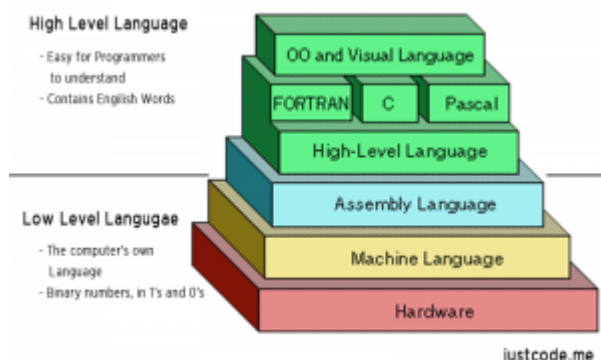


Bild X. Visar vad som ingår i Låg-resp. Hög-nivå av ett "språk".

**8b. Automatiserade DevOpsflöden**

DevOps-flödet brukar även kallas "DevOps Life Cycle". För att få alla som är involverade i ett utvecklingsprojekt att jobba mot ett och samma gemensamma mål tar "DevOps Life Cycle" hjälp av automatiserade lösningar för att bland annat planera, bygga, koda, testa och distribuera. Att Python är mångsidigt gör automatiseringen smidig och enkel.

**8c. Systemautomation**

Ett exempel på när systemautomation används är inom styrning av robotar, "Robotic Process Automation (RPA)". Fördelar med Python kontra många andra tillgängliga RPA plattformar är att Python ger bland annat stabilitet och tillåter körning av multipla processer på samma maskin. Det är vanligt att RPA använder sig av Artificiell intelligens (AI) och maskininlärning (ML). Även här är det en fördel för användning av Python då det ofta är det valda programmeringsspråket i dessa program.

**8d. Automatiserad testning**

Enligt källan [automationpanda](#) är Python ett av dom bästa valen vid automatiserade tester. Det nämns tio olika skäl till varför det anses vara så, se nedan.

**#1: Zen of Python** (för att läsa om riktlinjerna för Pythons PEP20 gå in på följande länk [PEP 20](#))

**#2: pytest**

"Pytest" är enligt källan det bästa testramverket som finns tillgängligt för programspråk. Testet hjälper till vid både enkla enhetstester och även komplexa funktionstester.

**#3: Paket**

Python har ett rikt bibliotek av användbara paket för att testning. Exempel pytest, unittest, logging, hypoteser. För att köra dess krävs endast en "pip-installation".

**#4: Multiparadigm**

Python är både objektorienterad och funktionell vilket är optimalt för testautomatisering.

**#5: "Typing Your Way"****#6: IDE: er(IDE, integrated development environment)**

För att få ett "språk" användarvänligt krävs ett gott IDE-stöd. Python stöder tex "JetBrains PyCharm visuell testning" med pytest samt unittest. Python tillägget som finns tillgängligt i Visual Studio Code är ett annat exempel.

**#7: Kommando-terminalens arbetsflöde**

Hela flödet av den automatiska testningen kan köras från kommando-terminalen.

**#8: Lätt att komma in**

Python är användarvänligt och passar både för nybörjare och expertprogrammerare. I den automatiserade världen är detta en stor fördel eftersom tester behöver utföras snabbt och enkelt.

**#9: Styrka för skalbarhet**

Kommando-terminalens mångsidighet gör att den passar in i alla arbetsflöden. Det innehåller många verktyg, är tydligt uppbyggt och med bra struktur. Alla dessa faktorer leder till att det bara finns en väg framåt, vilket är det rätta sättet .

**#10: Popularitet**

Python är det mest populära programmeringsspråket i världen, enligt källan.

**8e. Webserver Scripting**

Serverskript är en teknik som används vid utveckling av webbsidor. Script ligger på en server och på användarens begäran till den specifika webbsidan så kopplas skripten in och ger ett kundanpassat svar.

Python har ett verktyg, Scrapy som effektivt extraherar (scrape) ut information från webbplatser. Att manuellt ta ut information från flera webbsidor är tidskrävande. I stället är lösningen att göra detta automatiskt. Det brukar kallas att man använder sig av "spindlar" eller "sökrobotar" som snabbt och effektivt söker igenom webbsidor efter information.

**Källor:**

<https://medium.com/jay-tillu/difference-between-scripting-language-and-programming-language-4d520ebf36ef>  
<https://www.howtopython.org/is-python-a-programming-language-or-scripting-language/>  
<https://www.ssla.co.uk/low-level-developer/>  
<https://www.devopsuniversity.org/how-does-a-devops-engineer-work-with-python/>  
<https://www.activestate.com/blog/top-10-python-tools-to-make-your-life-easier/>  
<https://www.simplilearn.com/tutorials/rpa-tutorial/python-rpa>  
<https://automationpanda.com/2018/07/26/why-python-is-great-for-test-automation/>  
[https://en.wikipedia.org/wiki/Server-side\\_scripting](https://en.wikipedia.org/wiki/Server-side_scripting)