

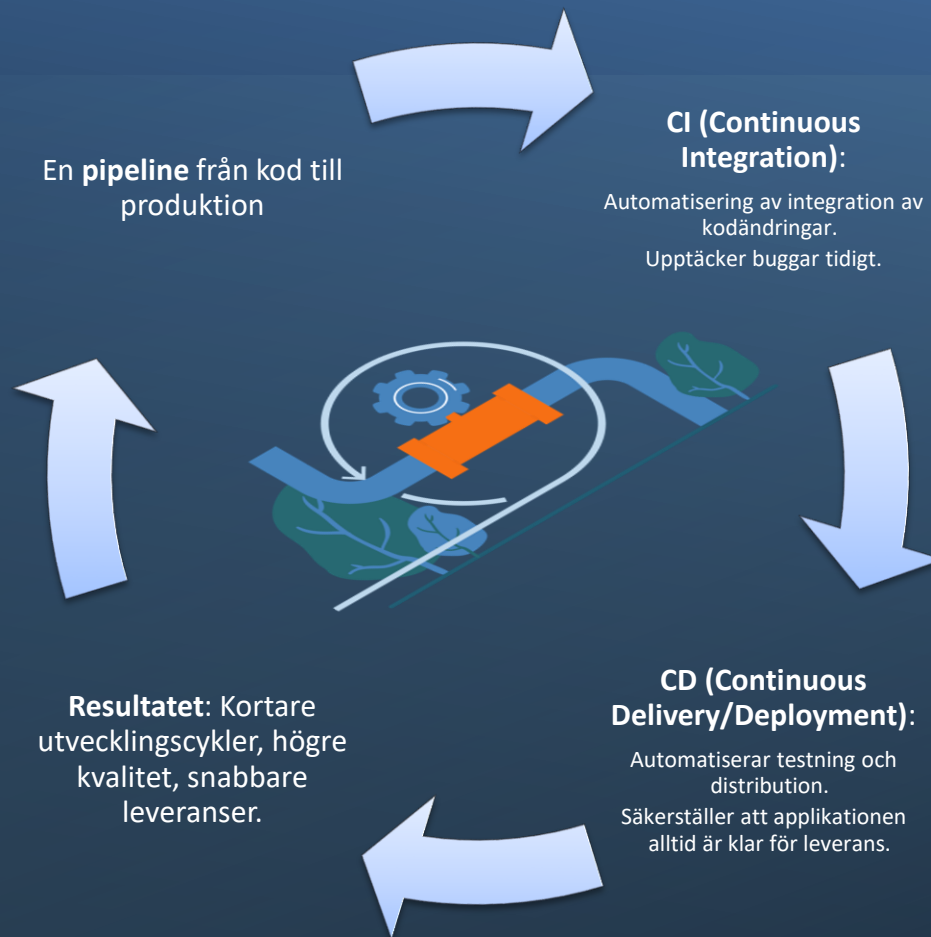
CI/CD & DevOps med GitHub Actions

Automatisering för
modern
mjukvaruutveckling

“It is not the strongest of the species, nor the most intelligent that survives. It is the one that is the most adaptable to change.”

C.D.

Vad är CI/CD?



Vad är CI/CD?

Continuous Integration (CI):

1. Automatisk byggning och testning av kodändringar.
2. Identifierar problem tidigt i utvecklingscykeln.
3. Säkerställer att ny kod fungerar med befintlig kodbas.

Continuous Deployment/Delivery (CD):

1. Automatiserar leveransen av ändringar.
2. Continuous Delivery: Till staging-miljö, manuellt beslut för produktion.
3. Continuous Deployment: Full automation till produktion.

Varför är CI/CD viktigt?

1. Snabbare tid till marknad
2. Förbättrad kvalitet
3. Bättre samarbete
4. Större agilitet
5. Kontinuerlig feedback
6. Bättre säkerhet
7. Kostnadsbesparingar
8. Konkurrensfördel



Komponenter i en CI/CD-pipeline



Bygghfas:

Automatisk kompilering av kod.



Testfas:

Automatiska enhetstester, integrationstester, säkerhetstester.



Distributionsfas:

Automatisk lansering till staging och/eller produktion.

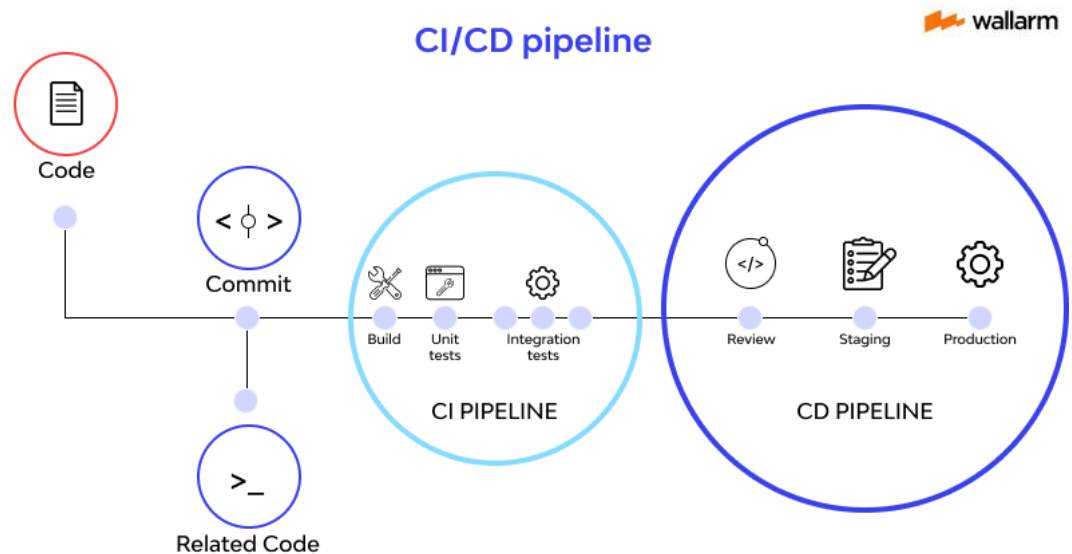
Hur fungerar CI/CD?

Automatiserad pipeline:

1. **Kodförändringar** (t.ex. en pull request på GitHub).
2. **Bygg** (kompilering och beroenden).
3. **Testning** (enhetstester, integrationstester).
4. **Distribution** (registry, produktion, staging).

Mätning och feedback:

Analysera resultat för förbättring.



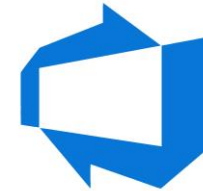
Vanliga fallgropar vid CI/CD-implementering

- a. Brist på testning
- b. Överdriven automatisering
- c. Dålig kommunikation
- d. Motstånd mot förändring
- e. Brist på standardisering
- f. Säkerhetsrisker



Hur börjar vi?

1. Starta smått:
 - Inför CI först, följt av CD.
2. Välj rätt verktyg:
 - Exempel: Azure DevOps, Jenkins, GitHub Actions.
3. Utbilda teamet:
 - Förståelse och stöd är avgörande.
4. Prioritera säkerhet:
 - Implementera säkerhetstestning tidigt i pipelinen.



Azure DevOps



Jenkins



GitHub Actions

```
# Option 1
{
  "hosting": "https://baz.bar",
  "api-key": "abcdefghijklmn12345",
  "attributes": ["email", "name"],
  "authentication": {
    "type": "IAM",
    "oauth": "https://oauth.baz.bar"
  }
}

# Option 2
hosting: https://baz.bar
api-key: abcdefghijklmn12345
attributes:
- email
- name
authentication:
  type: IAM
  oauth: https://oauth.baz.bar
```

CI/CD bidrar med

Högre kvalitet:

Upptäck fel tidigt genom automatiserade tester.

Snabbare leveranser:

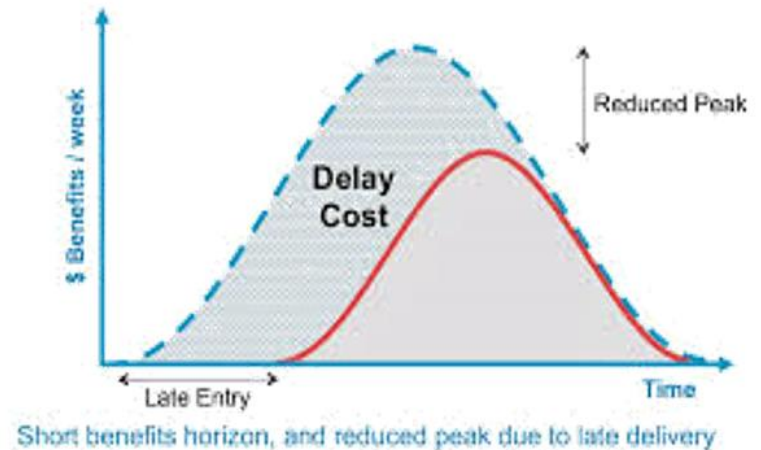
Distribution till produktion på några minuter.

Effektivitet:

Automatisering minskar manuell arbetsbörda.

Samarbete:

Förenklar arbetet i team med kontinuerlig kodintegration.



Impact of delayed release on benefits

Vad är DevOps och varför behövs det?

Definition:

Samarbete mellan utveckling (Dev) och drift (Ops) för snabb och stabil leverans av mjukvara.

Historiskt:

Utveckling och drift var separerade – vilket ledde till flaskhalsar.

Idag:

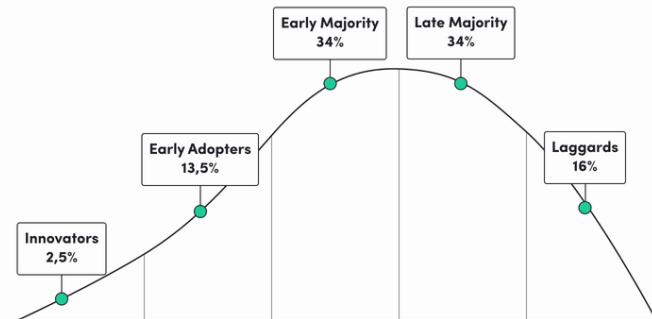
Krav på snabbare innovation och stabil drift.

Principer:

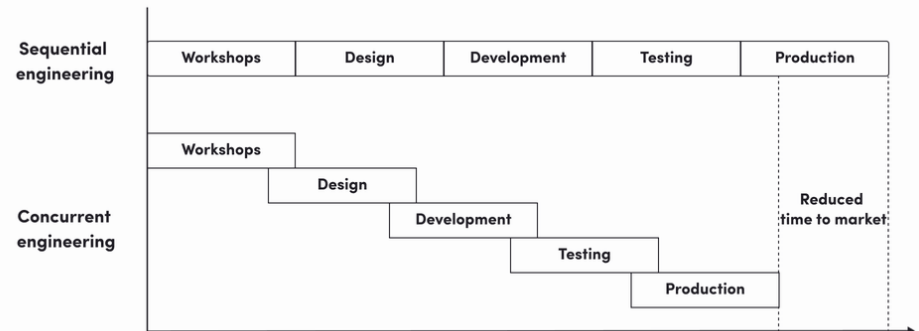
Kultur av samarbete.

Automatisering och mätbarhet.

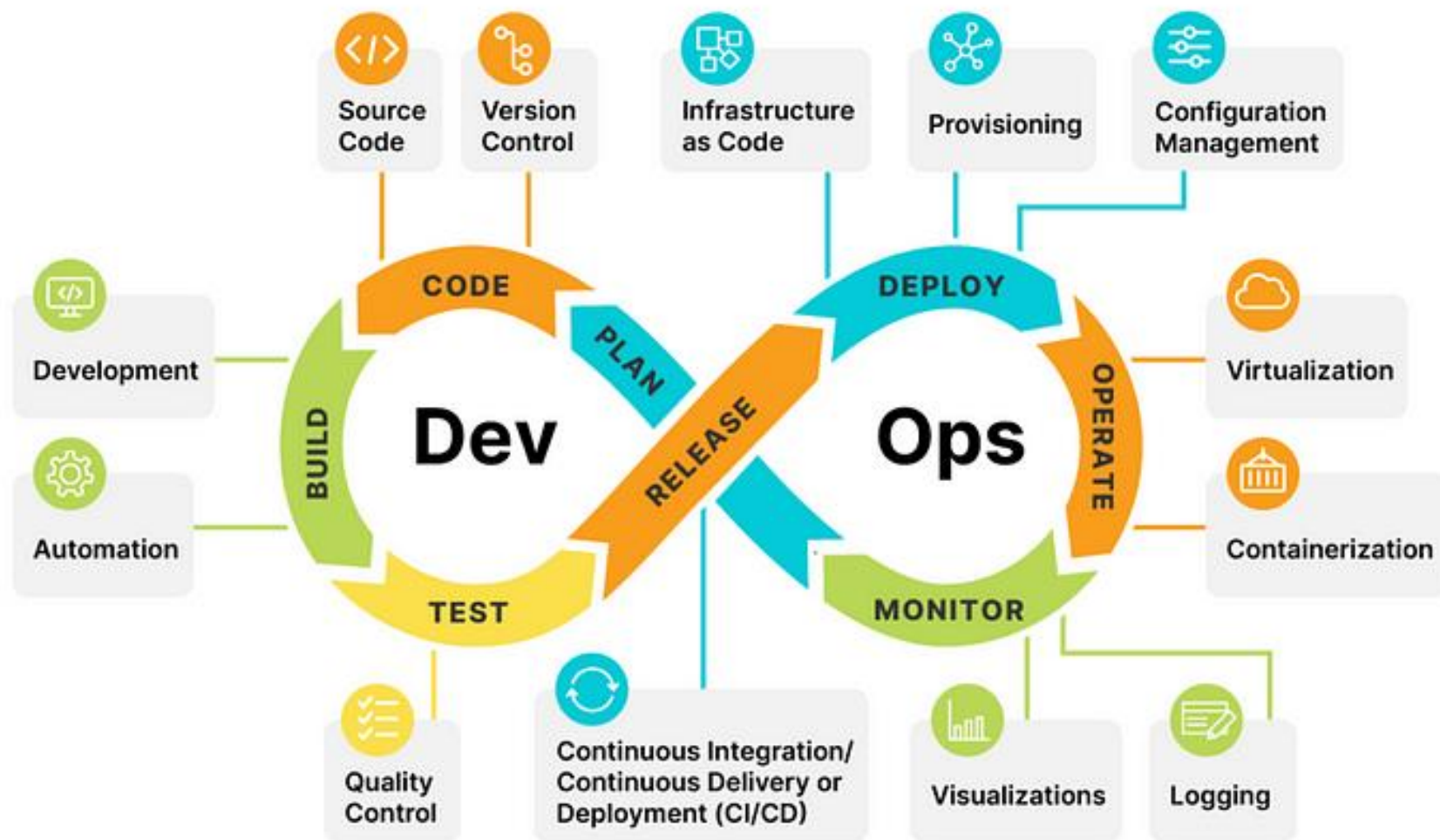
Product adoption curve



gorrion



gorrion



Introduktion till GitHub Actions

1. CI/CD-verktyg direkt integrerat i GitHub.
2. Workflows skrivs i YAML-format.
3. Automatisering för alla delar av DevOps-processen:
 - Bygg, testa, och distribuera kod.
 - Automatisera pull request-flöden.

Fördelar:

- Lätt att komma igång.
- Stor community och färdiga mallar.



CI/CD med GitHub Actions

Pipeline för en .NET-applikation:

- Trigger: Push till master.

- Steg:

Checkout kod.

Bygg projekt.

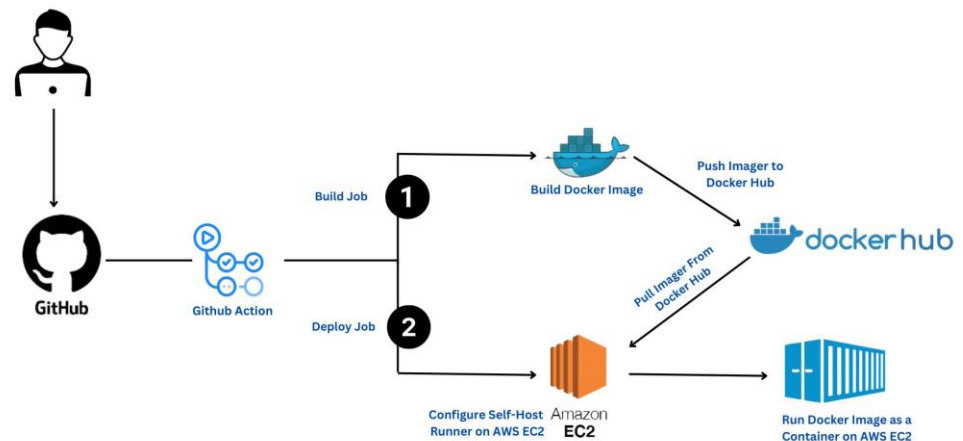
Skapa docker image.



...

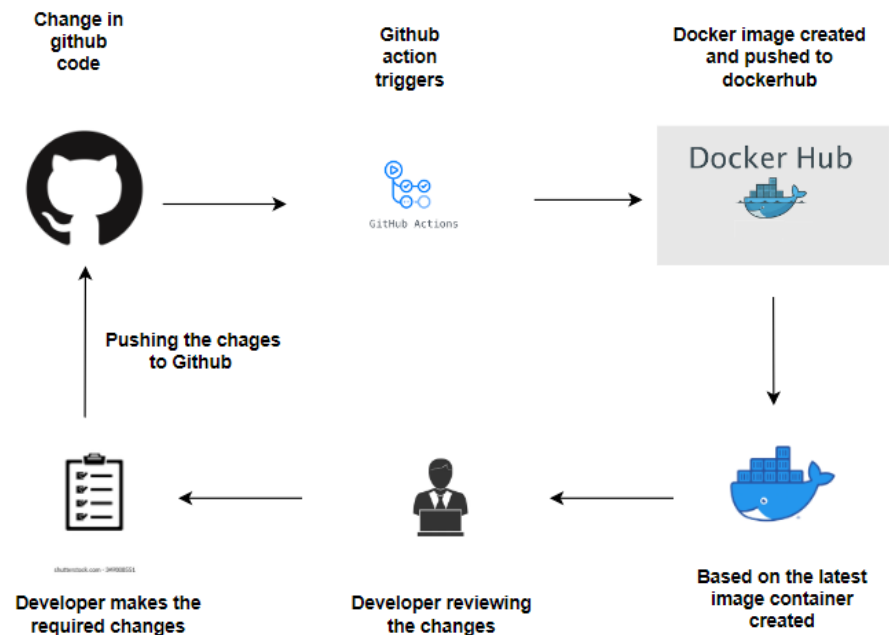
Deploy till Server.

- YAML-exempel tillämpas.



Fördelar med GitHub Actions

1. Enkel integration med GitHub-repo.
2. Flexibilitet att köra olika typer av workflow.
3. Open-source actions från communityn.
4. Kostnadseffektiv – gratis för små projekt.



Sammanfattning

- CI/CD är inte bara ett verktyg utan en kulturförändring.
- Automatisering möjliggör snabbare, säkrare och mer skalbar mjukvaruleverans.
- Undvik fallgropar genom planering, utbildning och standardisering.



CI/CD är nyckeln till effektiv mjukvaruutveckling.



DevOps förenar utveckling och drift för snabbare och stabilare leverans.



GitHub Actions gör CI/CD enkelt och flexibelt.



Börja med små pipelines och iterera för att förbättra.