

# STEPWISE IMPROVEMENT

---

Didaktisk princip i informatik

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

Læreren: *Har du gemt den sidste version, der fungerede?*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

Læreren: *Har du gemt den sidste version, der fungerede?*

Eleven: *Ja, det tror jeg nok*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

Læreren: *Har du gemt den sidste version, der fungerede?*

Eleven: *Ja, det tror jeg nok*

5 minutter senere

Eleven: *Jeg kan ikke finde den—hvad skal jeg gøre?*



# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

Læreren: *Har du gemt den sidste version, der fungerede?*

Eleven: *Ja, det tror jeg nok*

5 minutter senere

Eleven: *Jeg kan ikke finde den —hvad skal jeg gøre?*

Læreren: *Start forfra—og lav små ændringer og gem undervejs —gerne med versionsnumre*

# STEPWISE IMPROVEMENT

---

En typisk situation i undervisningen i informatik, når elever programmerer

Eleven: *Mit program fungerer ikke, som det skal—kan du ikke finde fejlen?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad var den sidste ændring, du lavede i dit program?*

Eleven: *Det ved jeg ikke lige, men programmet gider ikke at fungere*

Læreren: *Har du gemt den sidste version, der fungerede?*

Eleven: *Ja, det tror jeg nok*

5 minutter senere

Eleven: *Jeg kan ikke finde den —hvad skal jeg gøre?*

Læreren: *Start forfra—og lav små ændringer og gem undervejs —gerne med versionsnumre*

Eleven: *Jeg skal ud efter noget vand...*

# STEPWISE IMPROVEMENT

En typisk situation i undervisningen i informatik, når elever prøver

Eleven: *Mit program fungerer ikke, som det skal – hvad kan jeg gøre?*

Læreren: *Jo, det kan jeg sikkert godt, men hvad har du lavet i dit program?*

Eleven: *Det ved jeg ikke lige, men jeg har prøvet at gøre*

Læreren: *Har du gemt det?*

Eleven: *Ja, det har jeg.*

5 minutter

Eleven: *Jeg kan ikke finde den – hvad skal jeg gøre?*

Læreren: *Start forfra – og lav små ændringer og gem undervejs – gerne med versionsnumre*

Eleven: *Jeg skal ud efter noget vand...*

Programmering er kompliceret  
- gode vaner kan være en hjælp

# STEPWISE IMPROVEMENT

---

Idé

IT-system

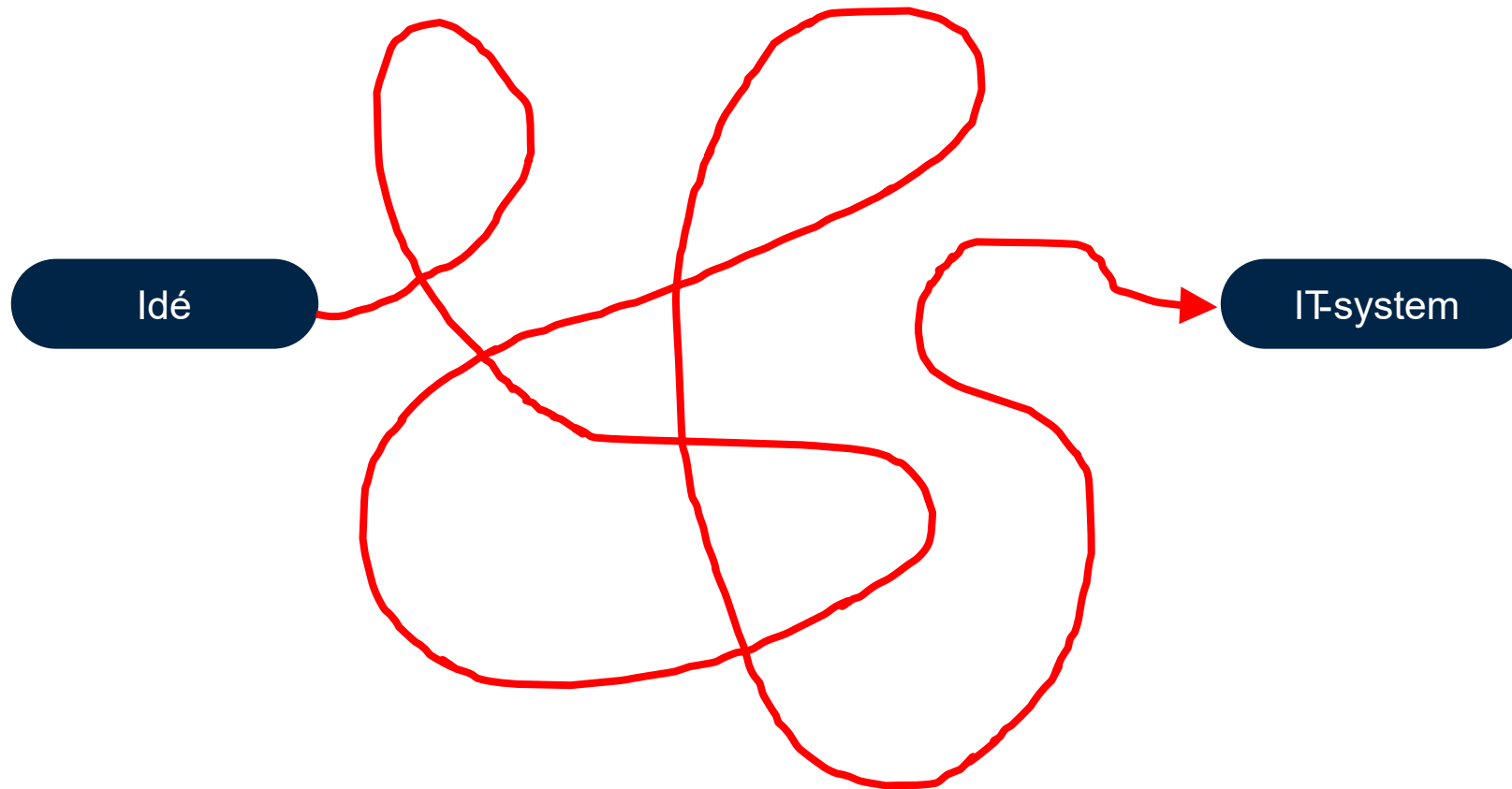
# STEPWISE IMPROVEMENT

---



# STEPWISE IMPROVEMENT

---



# STEPWISE IMPROVEMENT

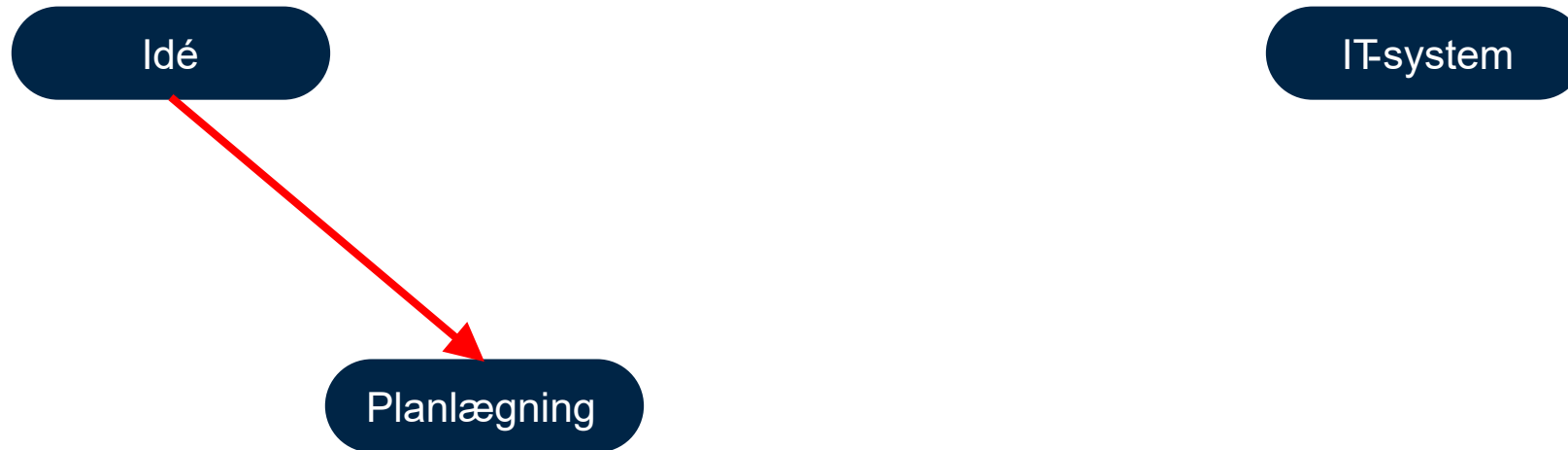
---

Idé

IT-system

# STEPWISE IMPROVEMENT

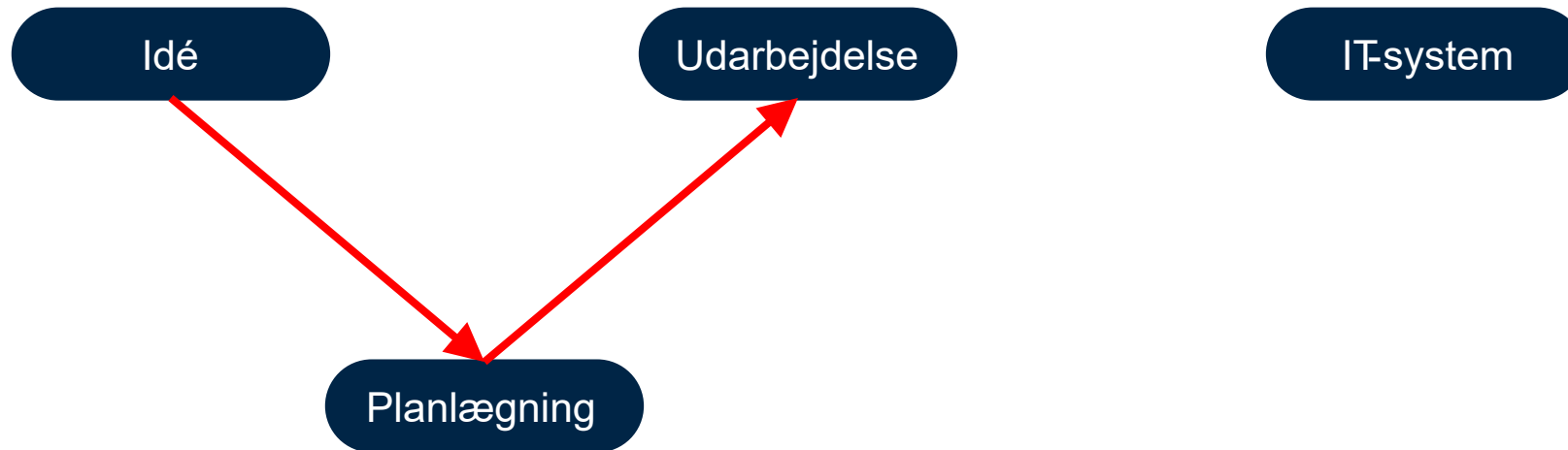
---





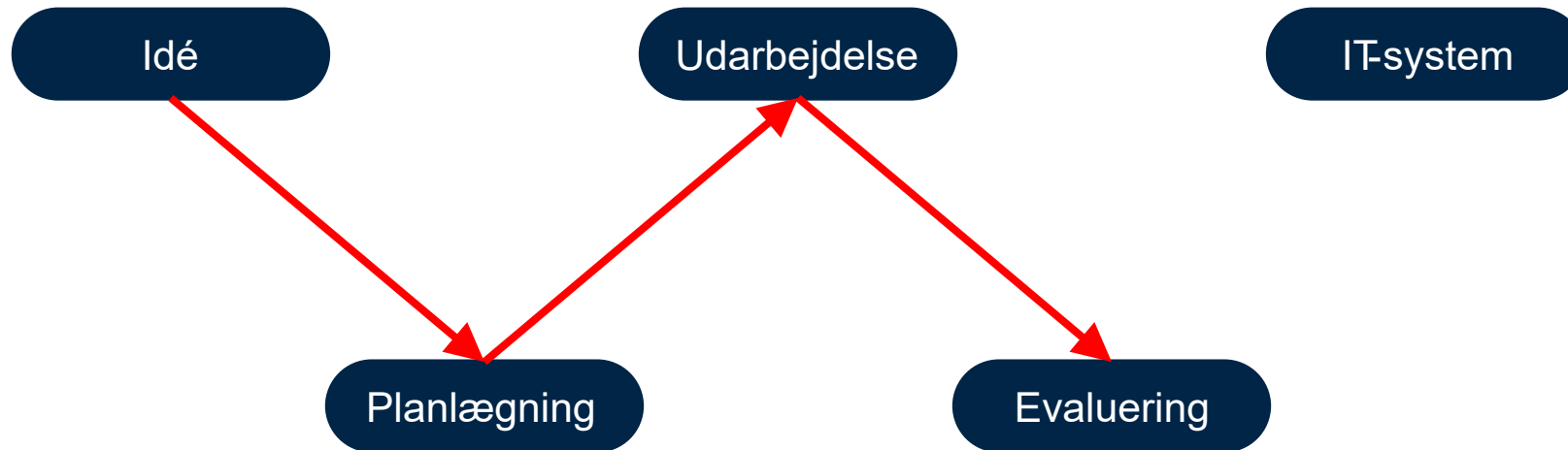
# STEPWISE IMPROVEMENT

---



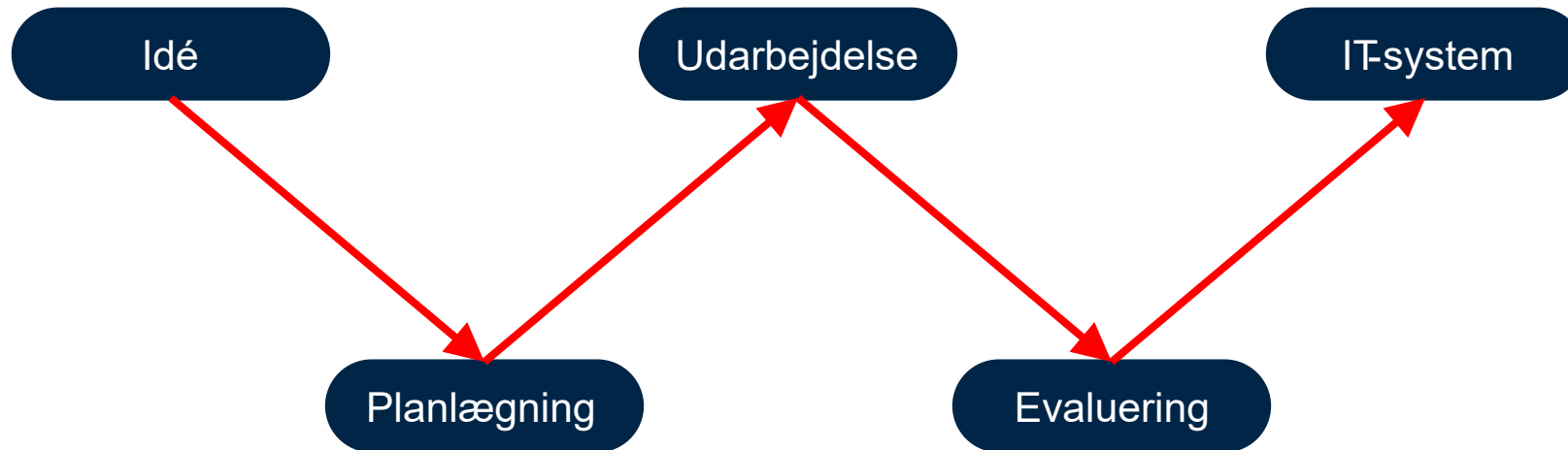
# STEPWISE IMPROVEMENT

---



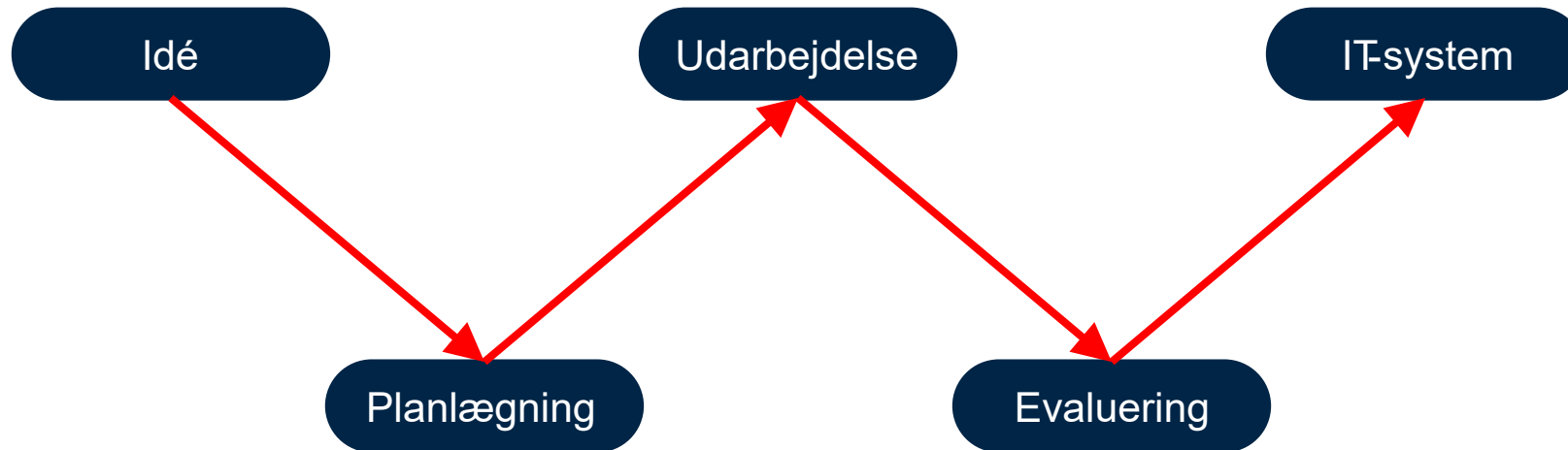
# STEPWISE IMPROVEMENT

---



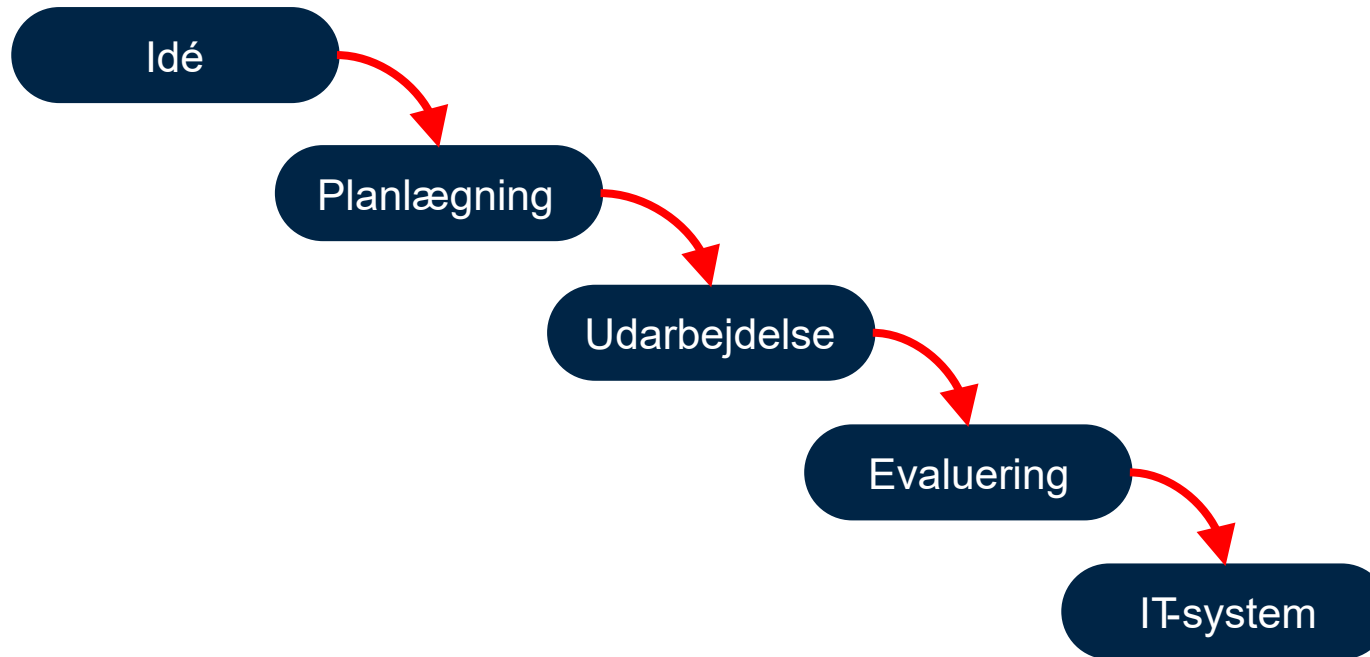
# STEPWISE IMPROVEMENT

Vandfaldsmetoden



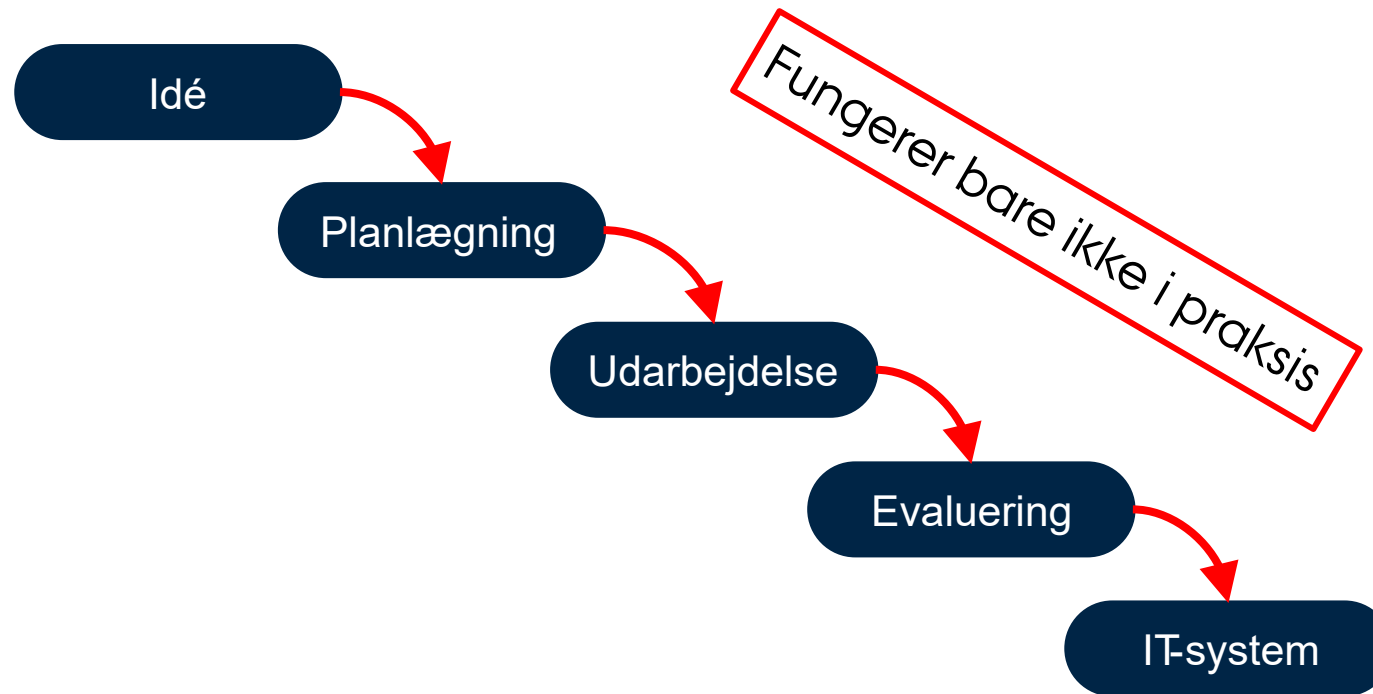
# STEPWISE IMPROVEMENT

Vandfaldsmetoden



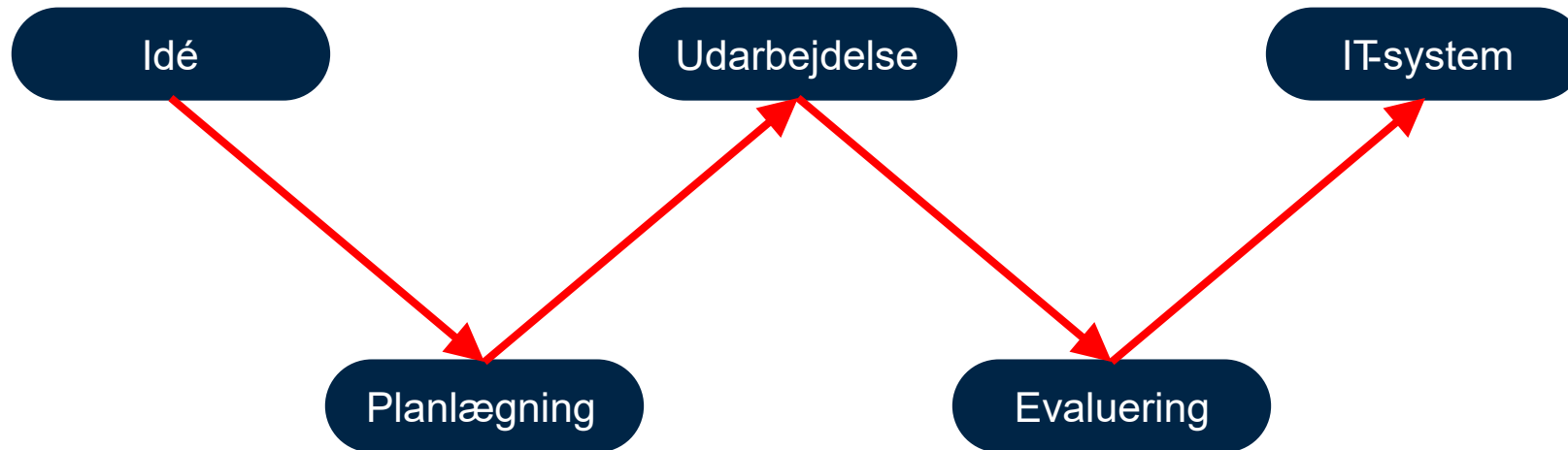
# STEPWISE IMPROVEMENT

Vandfaldsmetoden



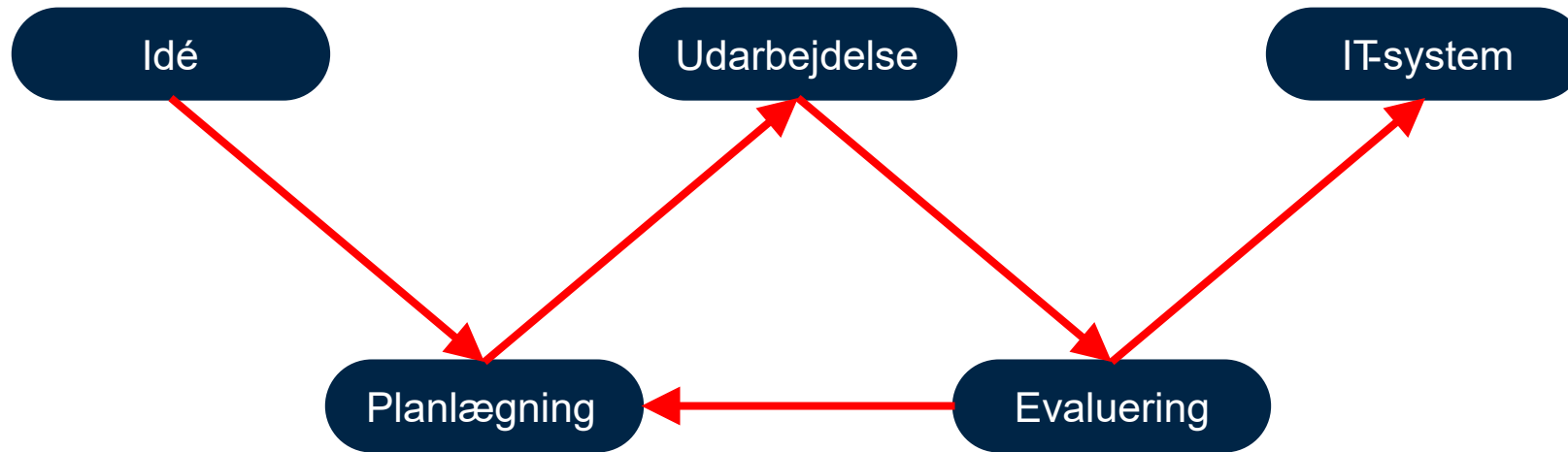
# STEPWISE IMPROVEMENT

---



# STEPWISE IMPROVEMENT

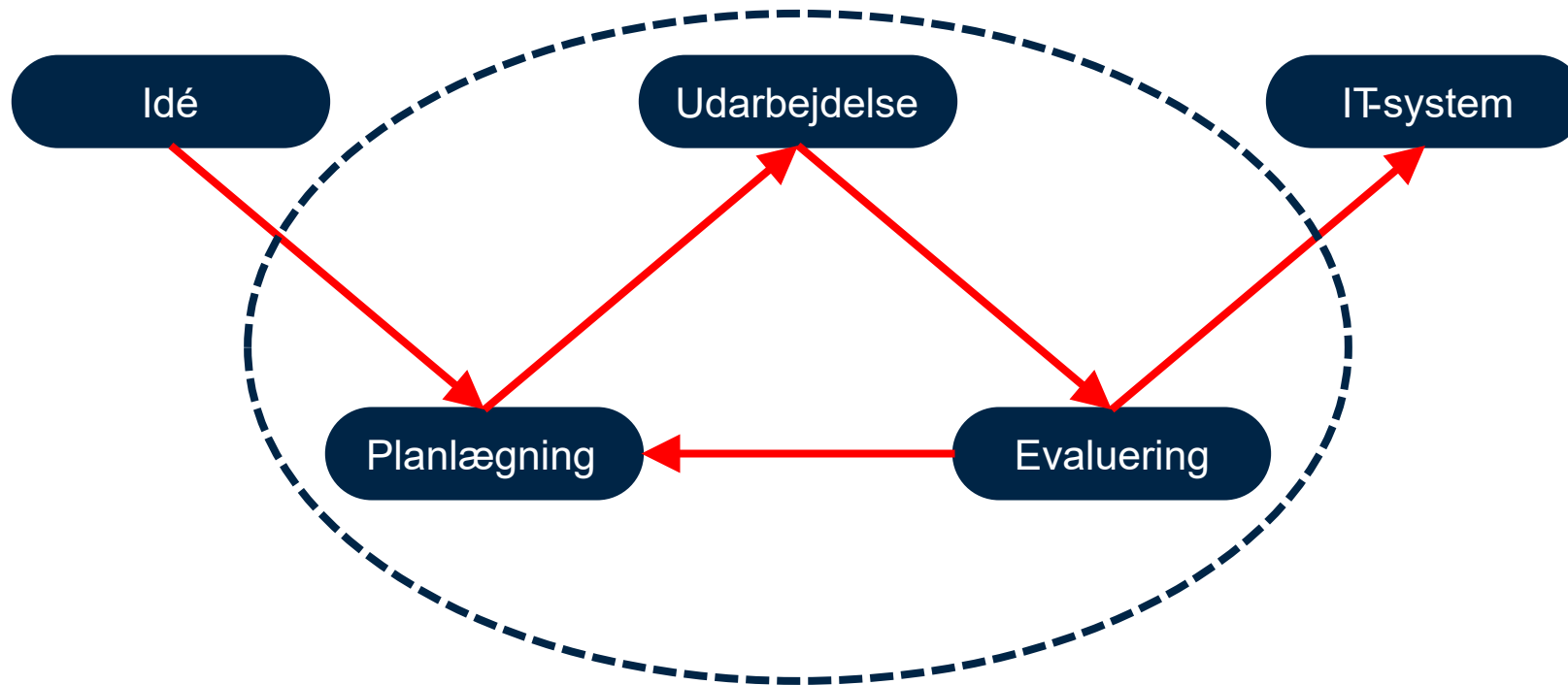
---





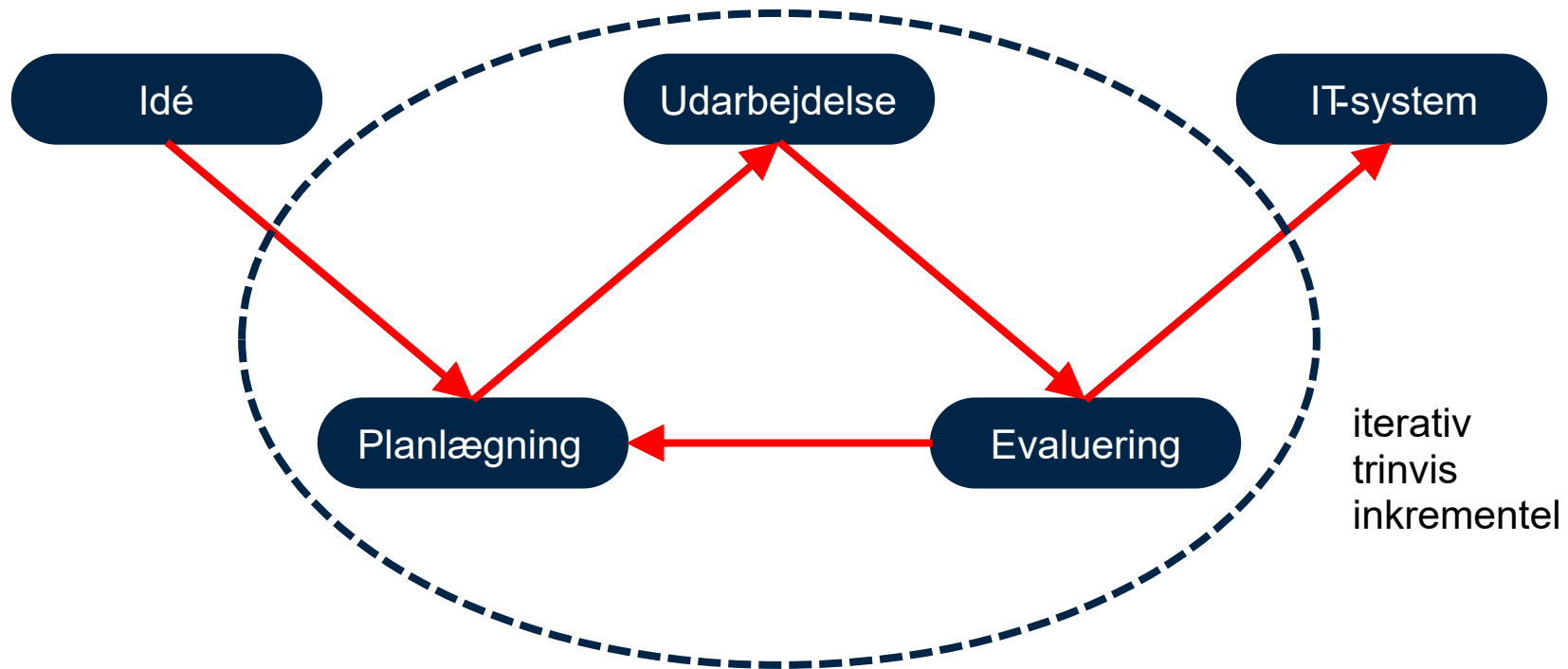
# STEPWISE IMPROVEMENT

---



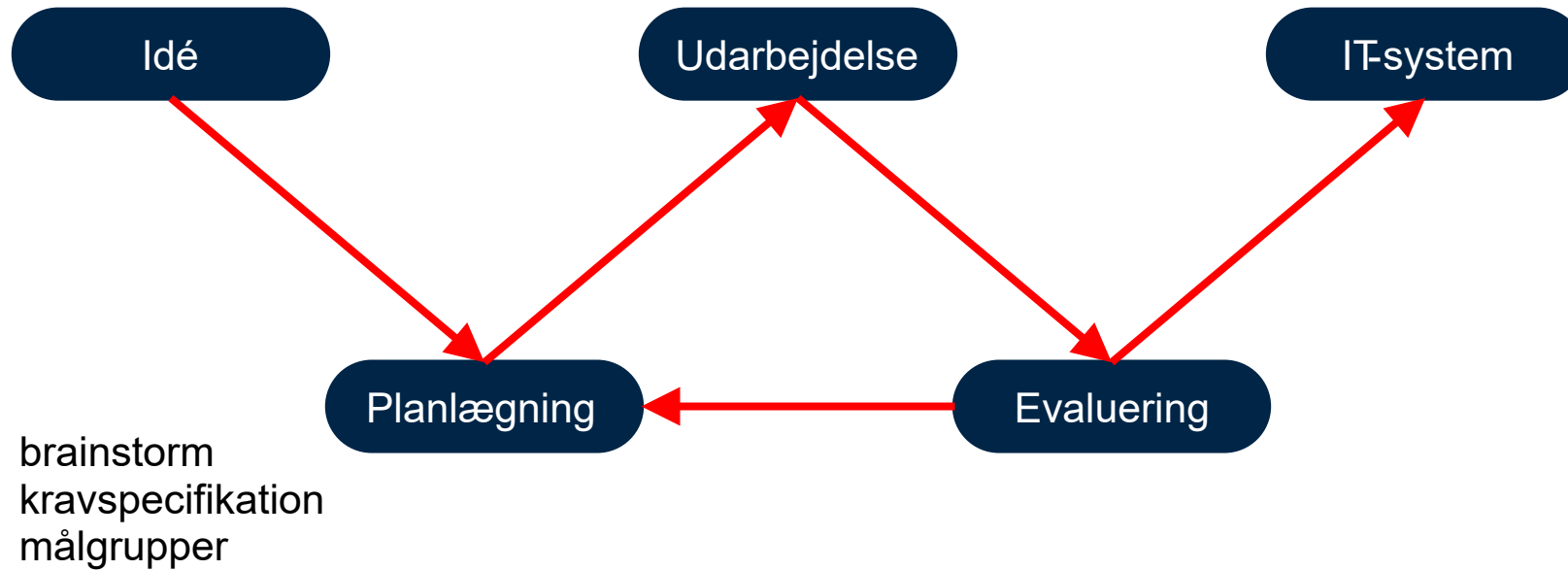
# STEPWISE IMPROVEMENT

---



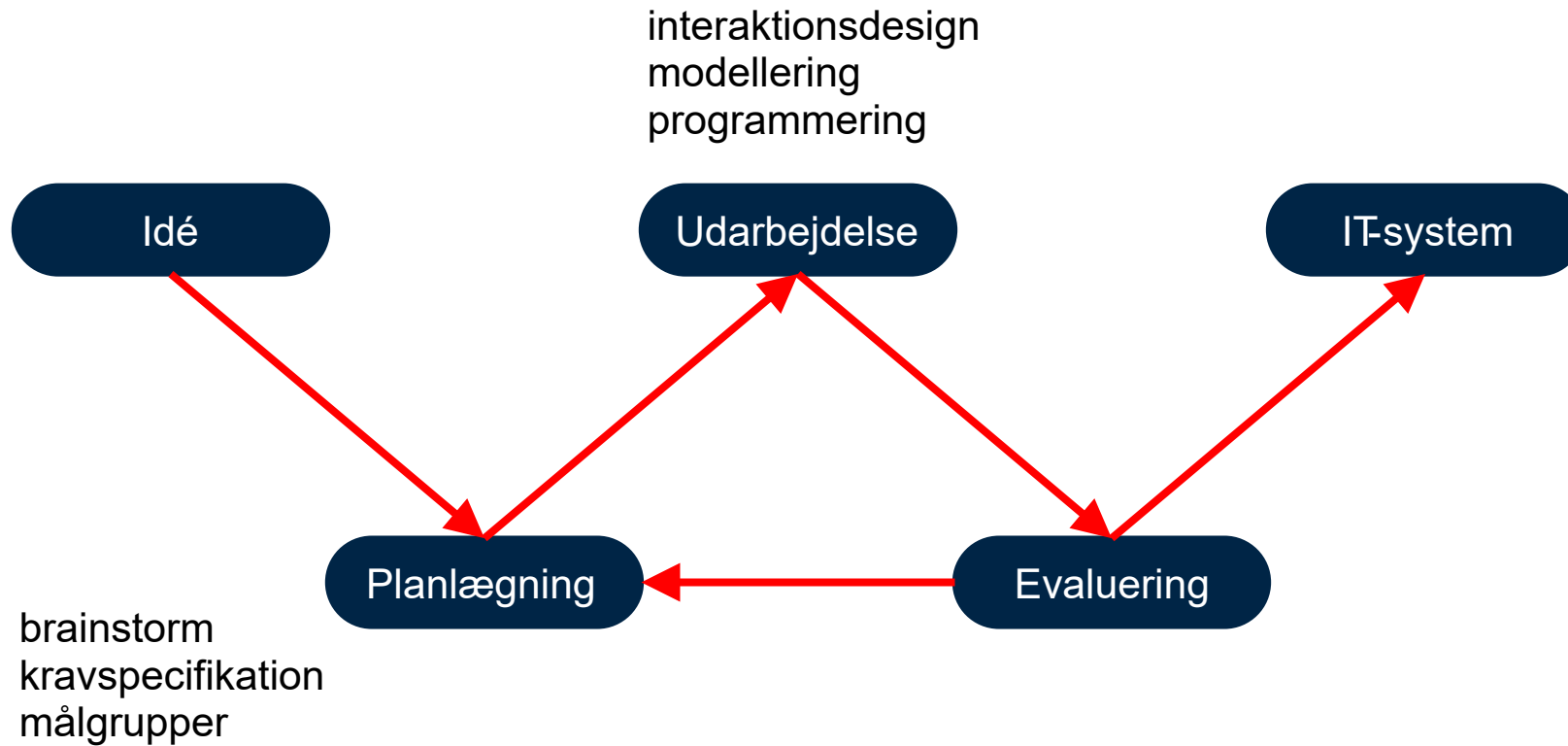
# STEPWISE IMPROVEMENT

---



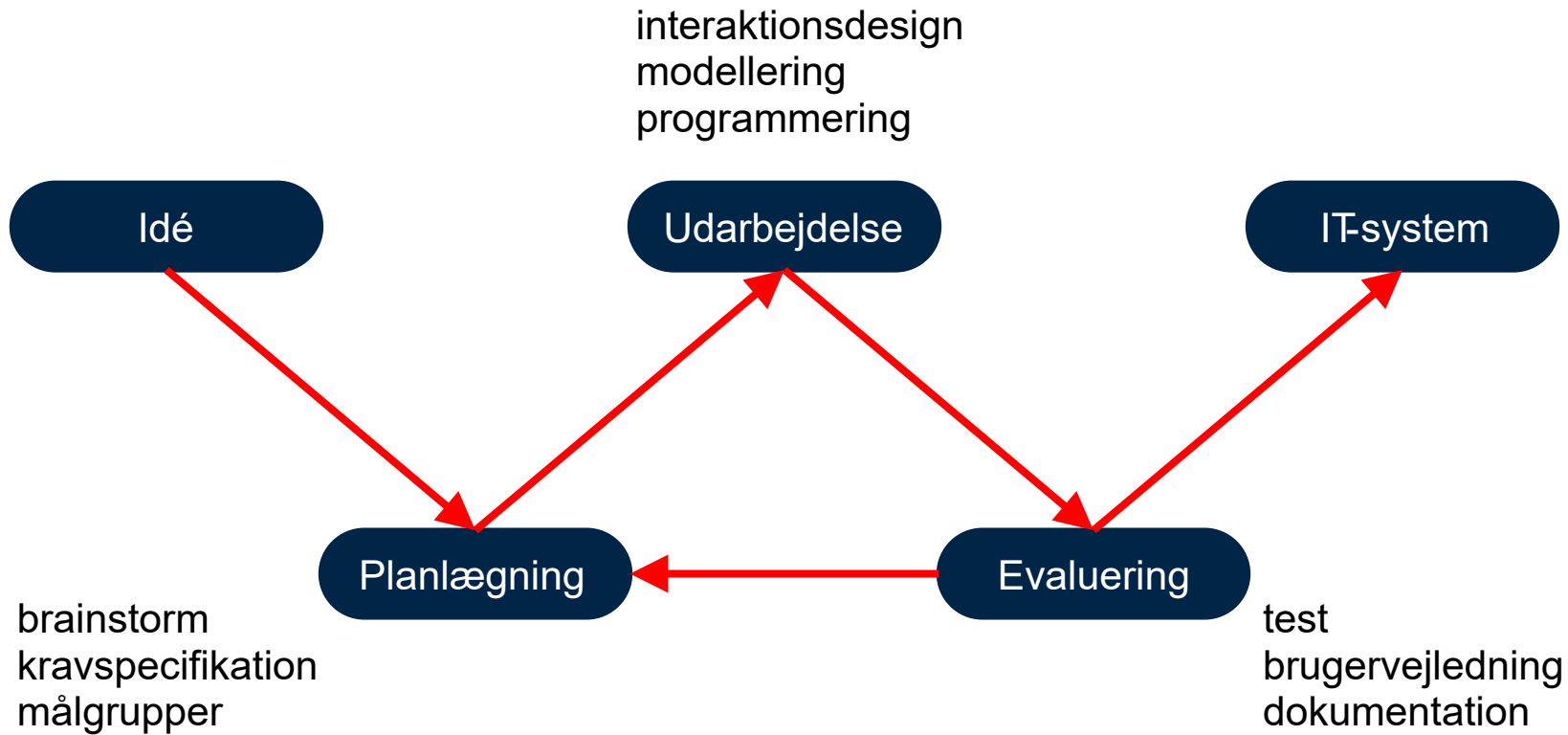
# STEPWISE IMPROVEMENT

---

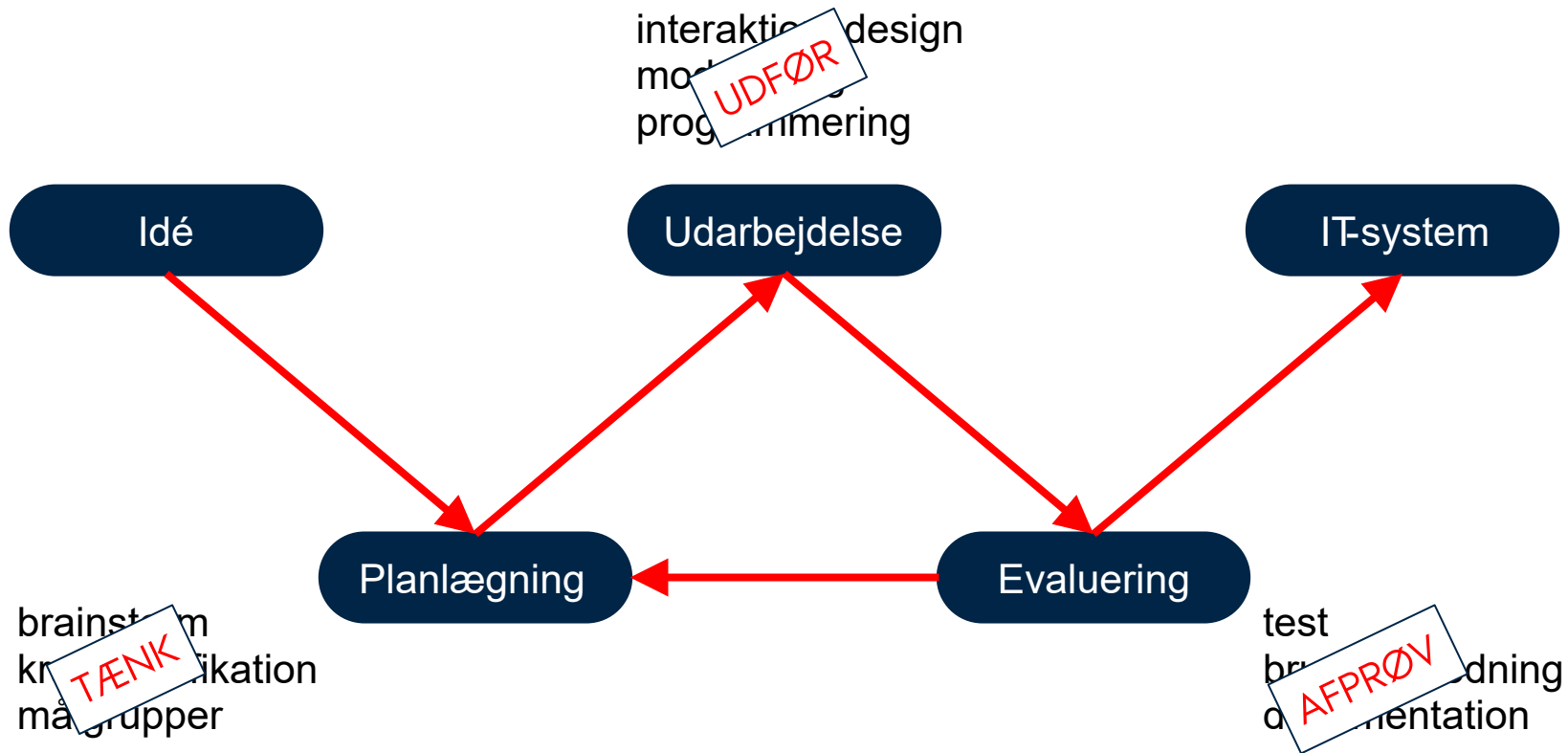


# STEPWISE IMPROVEMENT

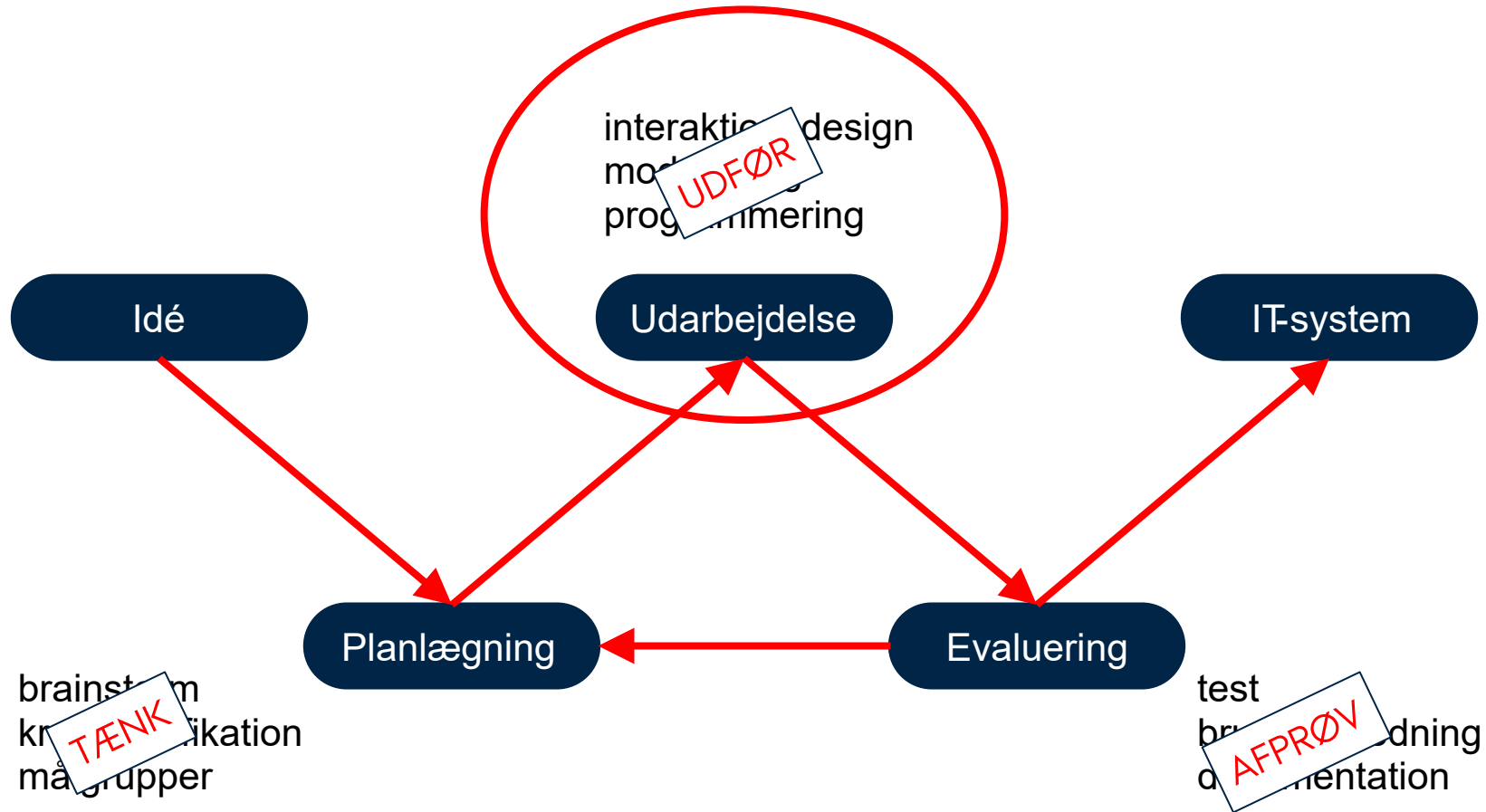
---



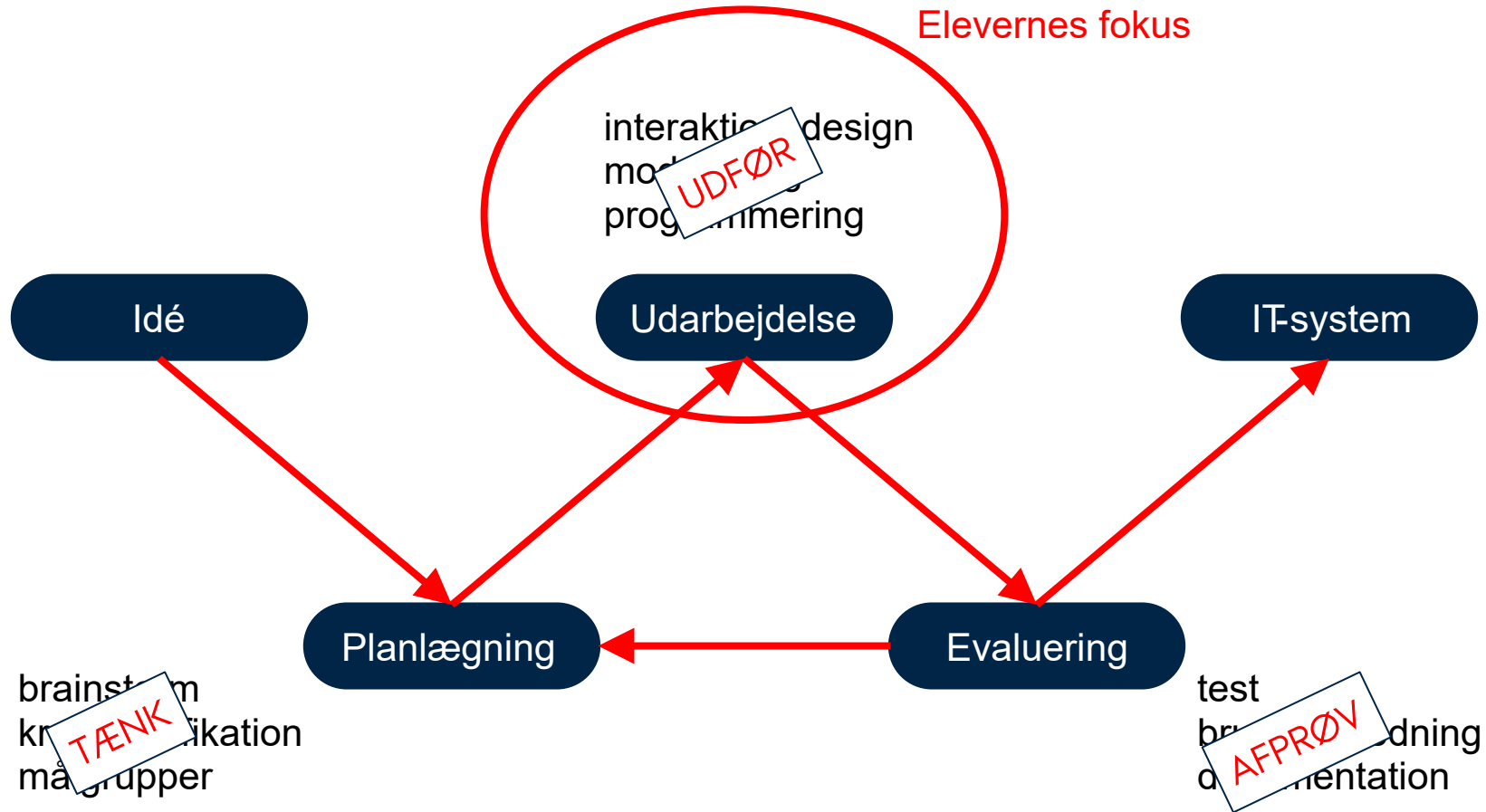
# STEPWISE IMPROVEMENT



# STEPWISE IMPROVEMENT



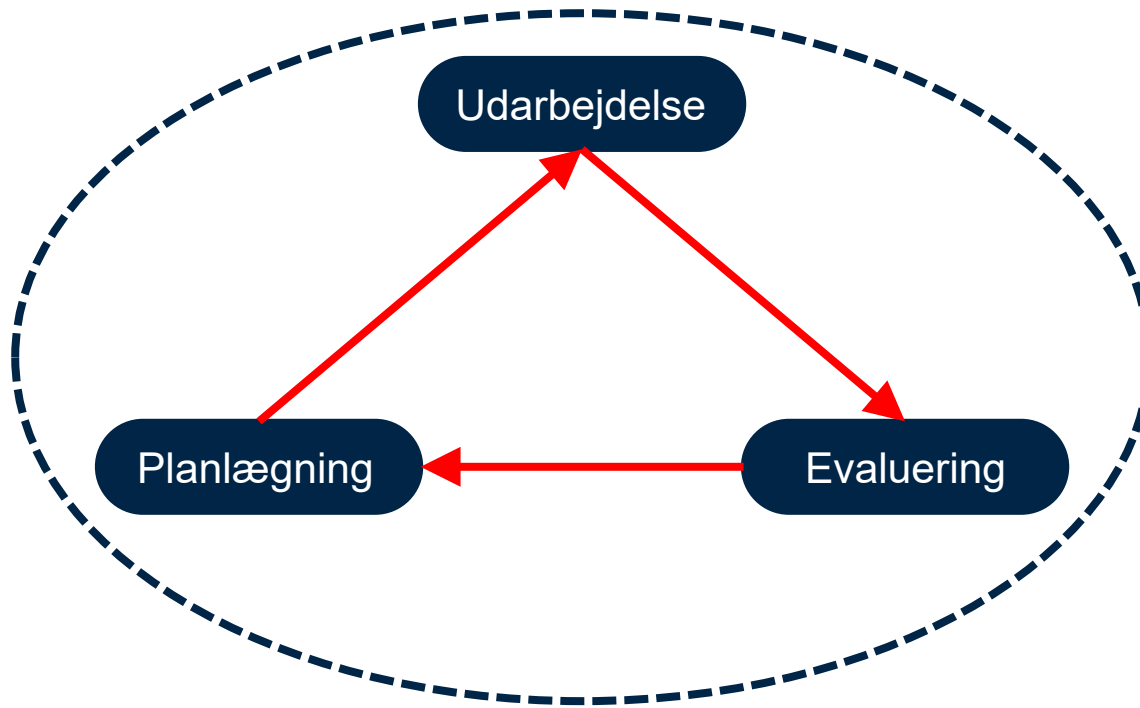
# STEPWISE IMPROVEMENT





# STEPWISE IMPROVEMENT

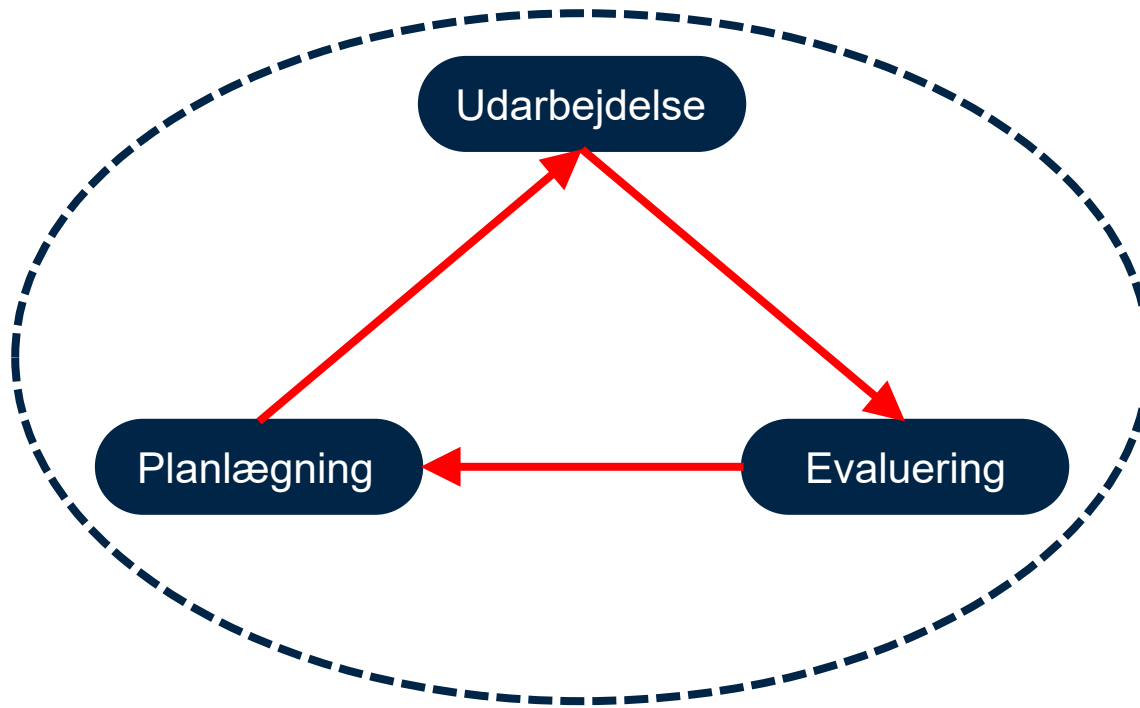
---



Udvikling foregår i tre dimensioner

# STEPWISE IMPROVEMENT

---

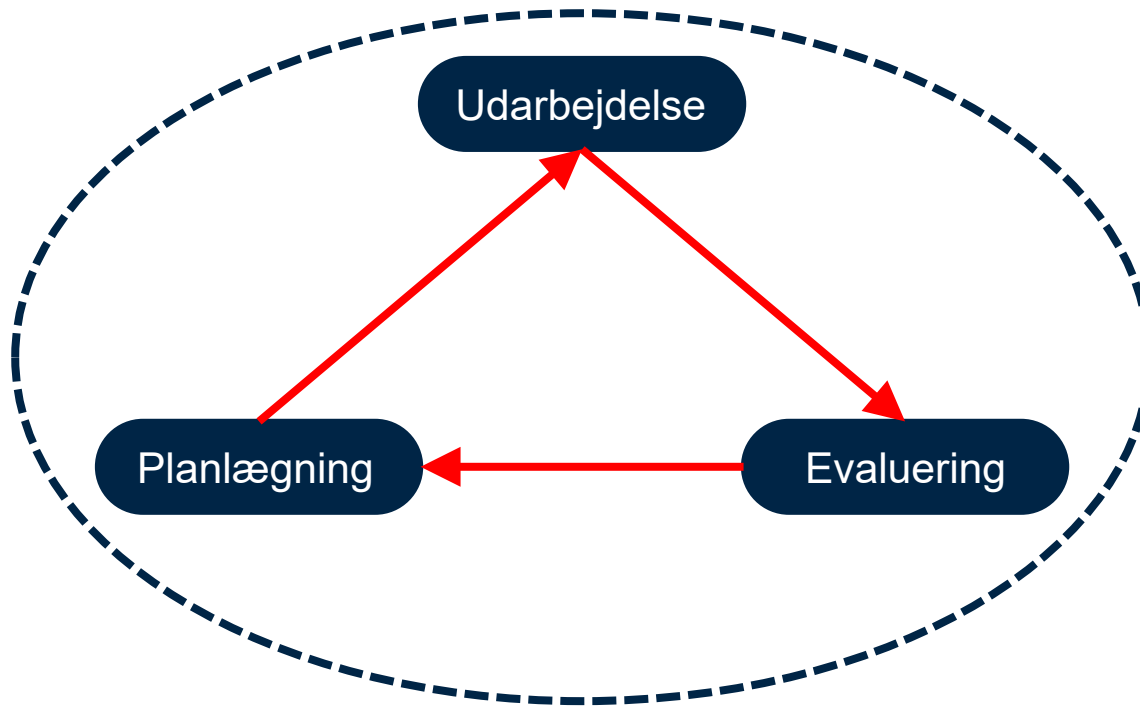


Udvikling foregår i tre dimensioner

- fra det abstrakte til det konkrete

# STEPWISE IMPROVEMENT

---

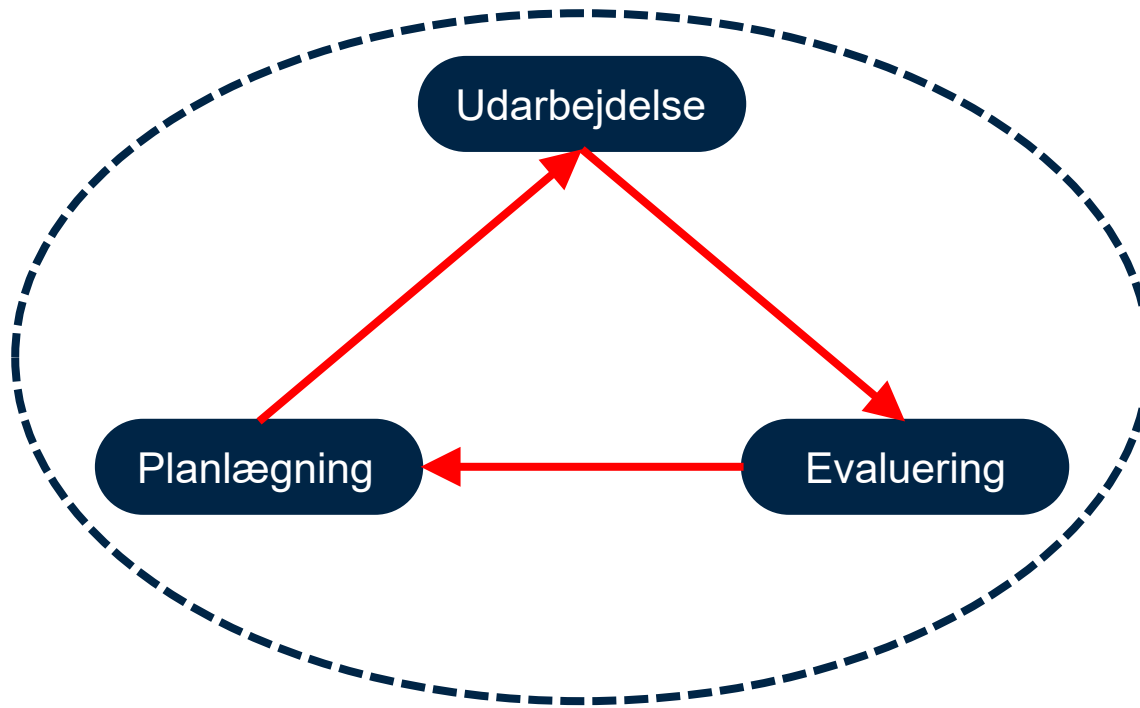


Udvikling foregår i tre dimensioner

- fra det abstrakte til det konkrete
- fra dele til helheder

# STEPWISE IMPROVEMENT

---



Udvikling foregår i tre dimensioner

- fra det abstrakte til det konkrete
- fra dele til helheder
- fra det ustrukturerede til det strukturerede

# STEPWISE IMPROVEMENT

---

Fra det abstrakte til det konkrete

- top-down design

# STEPWISE IMPROVEMENT

---

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)

# STEPWISE IMPROVEMENT

---

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Fra pseudokode til Pythonkode

# STEPWISE IMPROVEMENT

---

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Bestem det største tal af to tal



# STEPWISE IMPROVEMENT

---

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Bestem det største tal af to tal

`maks(tal1, tal2)`

# STEPWISE IMPROVEMENT

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Bestem det største tal af to tal

maks(tal1, tal2)



```
hvis tal1 > tal2
    returner tal1
ellers
    returner tal2
```

# STEPWISE IMPROVEMENT

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Bestem det største tal af to tal

maks(tal1, tal2)



```
hvis tal1 > tal2
    returner tal1
ellers
    returner tal2
```



```
if tal1 > tal2:
    return tal1
else:
    return tal2
```

# STEPWISE IMPROVEMENT

Fra det abstrakte til det konkrete

- top-down design
- konkretisering, forfining (= refinement)
- samme kravsspecifikation

Bestem det største tal af to tal

maks(tal1, tal2)



```
hvis tal1 > tal2
    returner tal1
ellers
    returner tal2
```



```
if tal1 > tal2:
    return tal1
else:
    return tal2
```



# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart



# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

Kalenderprogram, der kan bestemme den foregående dato og den efterfølgende

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

Kalenderprogram, der kan bestemme den foregående dato og den efterfølgende

- alle måneder har 30 dage

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

Kalenderprogram, der kan bestemme den foregående dato og den efterfølgende

- alle måneder har 30 dage → måneder har 31, 28, 31... dage

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

Kalenderprogram, der kan bestemme den foregående dato og den efterfølgende

- alle måneder har 30 dage ➡ måneder har 31, 28, 31... dage ➡ skudår hvert 4. år

# STEPWISE IMPROVEMENT

---

Fra dele til helheder

- bottom-up design
- udvidelse (= extension)
- kravsspecifikation udvides
- flere tilfælde –skalérbart

Algoritmisk kunst

- fra tegning af trekant til n-kant

Kalenderprogram, der kan bestemme den foregående dato og den efterfølgende

- alle måneder har 30 dage ➡ måneder har 31, 28, 31... dage ➡ skudår hvert 4. år ➡ ...

# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)

# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten



# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten
- fjerne redundant kode

# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten
- fjerne redundant kode
- forbedre læsbarheden

# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten
- fjerne redundant kode
- forbedre læsbarheden
- lave moduler

# STEPWISE IMPROVEMENT

---

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten
- fjerne redundant kode
- forbedre læsbarheden
- lave moduler
- optimere algoritmen

# STEPWISE IMPROVEMENT

Fra det ustrukturerede til det strukturerede

- omstrukturering (= restructuring)
- omskrivning af kode uden at ændre på funktionaliteten
- fjerne redundant kode
- forbedre læsbarheden
- lave moduler
- optimere algoritmen

```
In [3]: a = 5  
        b = 7  
        temp = a  
        a = b  
        b = temp
```

```
In [4]: a, b
```

```
Out[4]: (7, 5)
```



```
In [5]: a = 5  
        b = 7  
        a, b = b, a
```

```
In [6]: a, b
```

```
Out[6]: (7, 5)
```

# STEPWISE IMPROVEMENT

---

Udvikling foregår i tre dimensioner

# STEPWISE IMPROVEMENT

—  
Udvikling foregår i tre dimensioner

En iteration foregår i én af tre

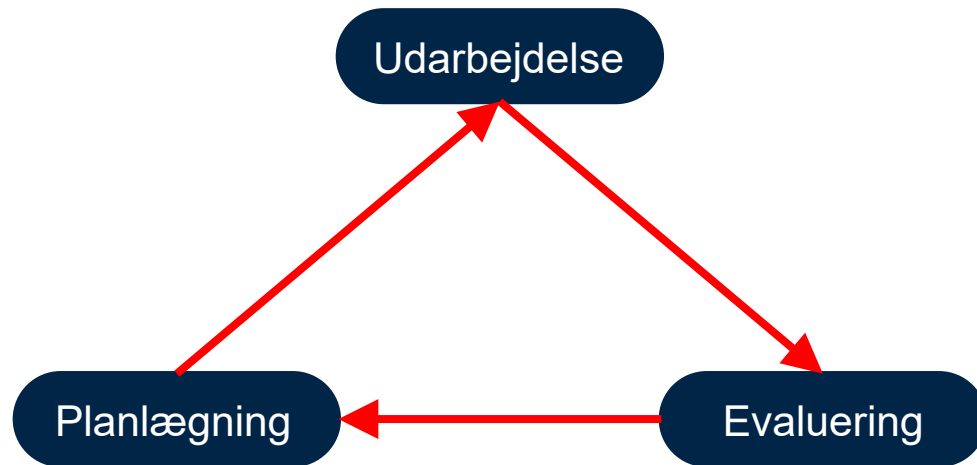
- fra det abstrakte til det konkrete (= **refine**)
- fra dele til helheder (= **extend**)
- fra det ustrukturerede til det strukturerede (= **restructure**)

# STEPWISE IMPROVEMENT

Udvikling foregår i tre dimensioner

En iteration foregår i én af tre

- fra det abstrakte til det konkrete (= **refine**)
- fra dele til helheder (= **extend**)
- fra det ustrukturerede til det strukturerede (= **restructure**)



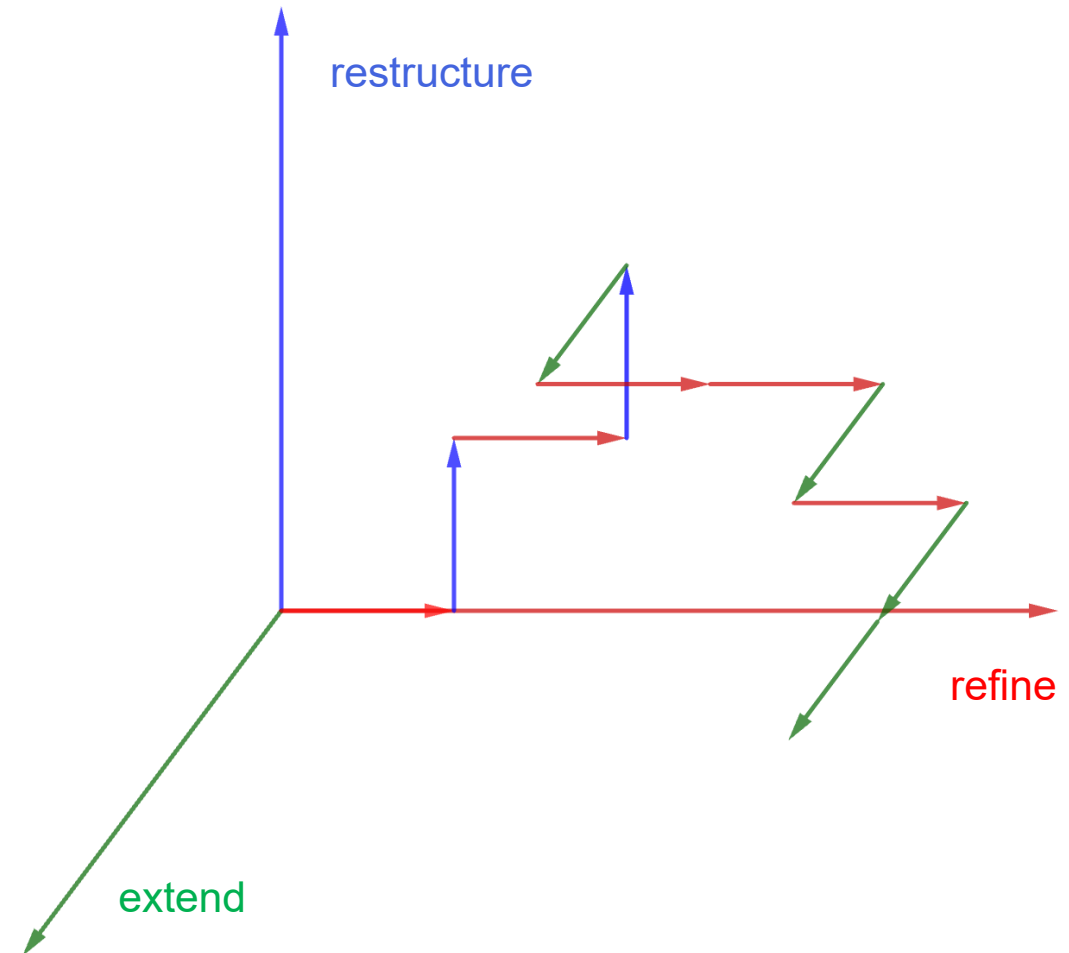
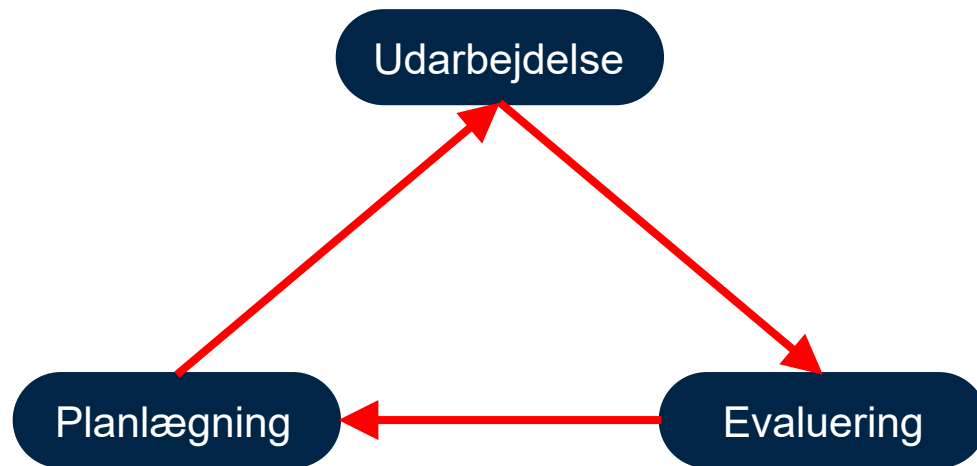


# STEPWISE IMPROVEMENT

Udvikling foregår i tre dimensioner

En iteration foregår i én af tre

- fra det abstrakte til det konkrete (= **refine**)
- fra dele til helheder (= **extend**)
- fra det ustrukturerede til det strukturerede (= **restructure**)



# OPSUMMERING

---

*Stepwise Improvement* er en disciplinerende metode, der

- sikrer en god og velstruktureret måde at udvikle digitale artefakter på

også

# OPSUMMERING

---

*Stepwise Improvement* er en disciplinerende metode, der

- sikrer en god og velstruktureret måde at udvikle digitale artefakter på
- sikrer en trinvis udvikling af det digitale artefakt

også

# OPSUMMERING

---

*Stepwise Improvement* er en disciplinerende metode, der

- sikrer en god og velstruktureret måde at udvikle digitale artefakter på
- sikrer en trinvis udvikling af det digitale artefakt
- mindsker arbejdet med fejlfinding ('debugging')

også

# OPSUMMERING

---

*Stepwise Improvement* er en disciplinerende metode, der

- sikrer en god og velstruktureret måde at udvikle digitale artefakter på
- sikrer en trinvis udvikling af det digitale artefakt
- mindsker arbejdet med fejlfinding ('debugging')

Metoden *Stepwise Improvement* kaldes også for *Den iterative arbejdsproces*

også

# OPSUMMERING

---

*Stepwise Improvement* er en disciplinerende metode, der

- sikrer en god og velstruktureret måde at udvikle digitale artefakter på
- sikrer en trinvis udvikling af det digitale artefakt
- mindsker arbejdet med fejlfinding ('debugging')

Metoden *Stepwise Improvement* kaldes også for *Den iterative arbejdsproces*

Metoden er vigtig for undervisere, når der skal planlægges undervisningsforløb, men den er også vigtige for elever, der skal lære at programmere