XLENT
CONSULTING GROUP

Koddesign

Hur kan designmönster hjälpa fundamentala designproblem?

Peter Starefeldt

XLENT
CONSULTING GROUP

Bra design?

- **Synbart när ändringar behöver genomföras**

XLENT
CONSULTING GROUP

Bra design?

- Synbart när ändringar behöver genomföras
- **Synbart när människor läser kod**

XLENT
CONSULTING GROUP

Bra design?

- Synbart när ändringar behöver genomföras
- Synbart när människor läser kod
- **Systemdelar är isolerade från varandra**

XLENT
CONSULTING GROUP

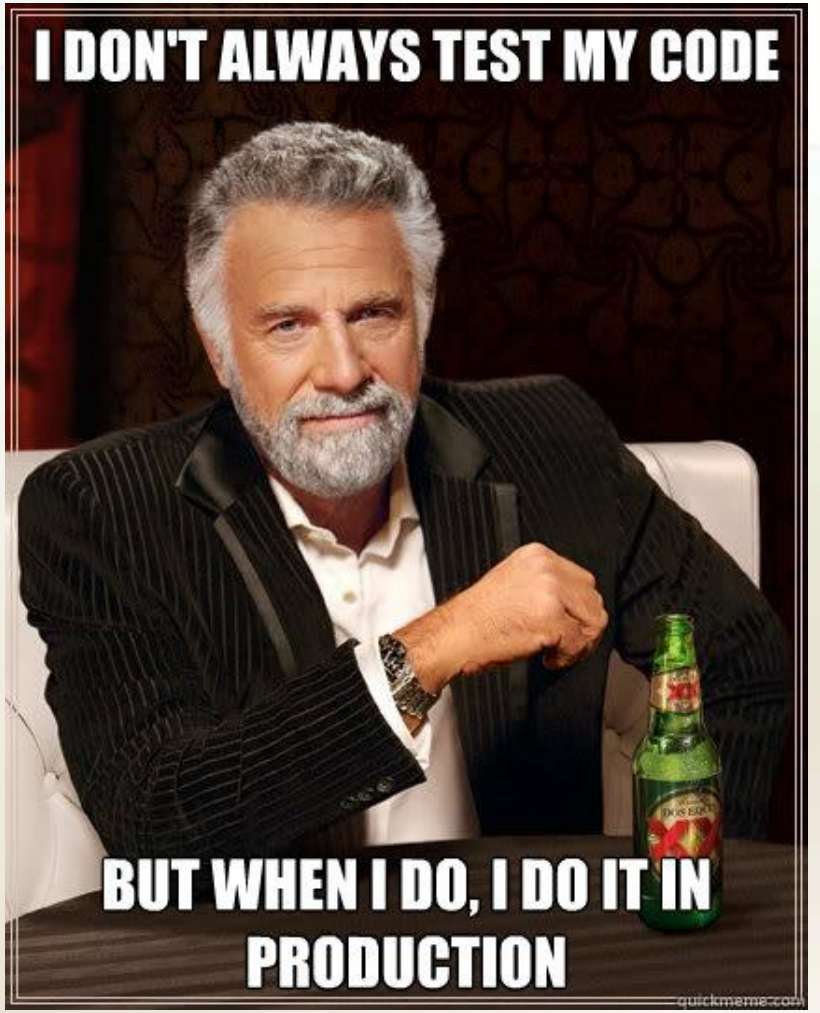
Bra design?

- Synbart när ändringar behöver genomföras
- Synbart när människor läser kod
- Systemdelar är isolerade från varandra
- **Grunden är oerhört betydelsefull**

XLENT CONSULTING GROUP

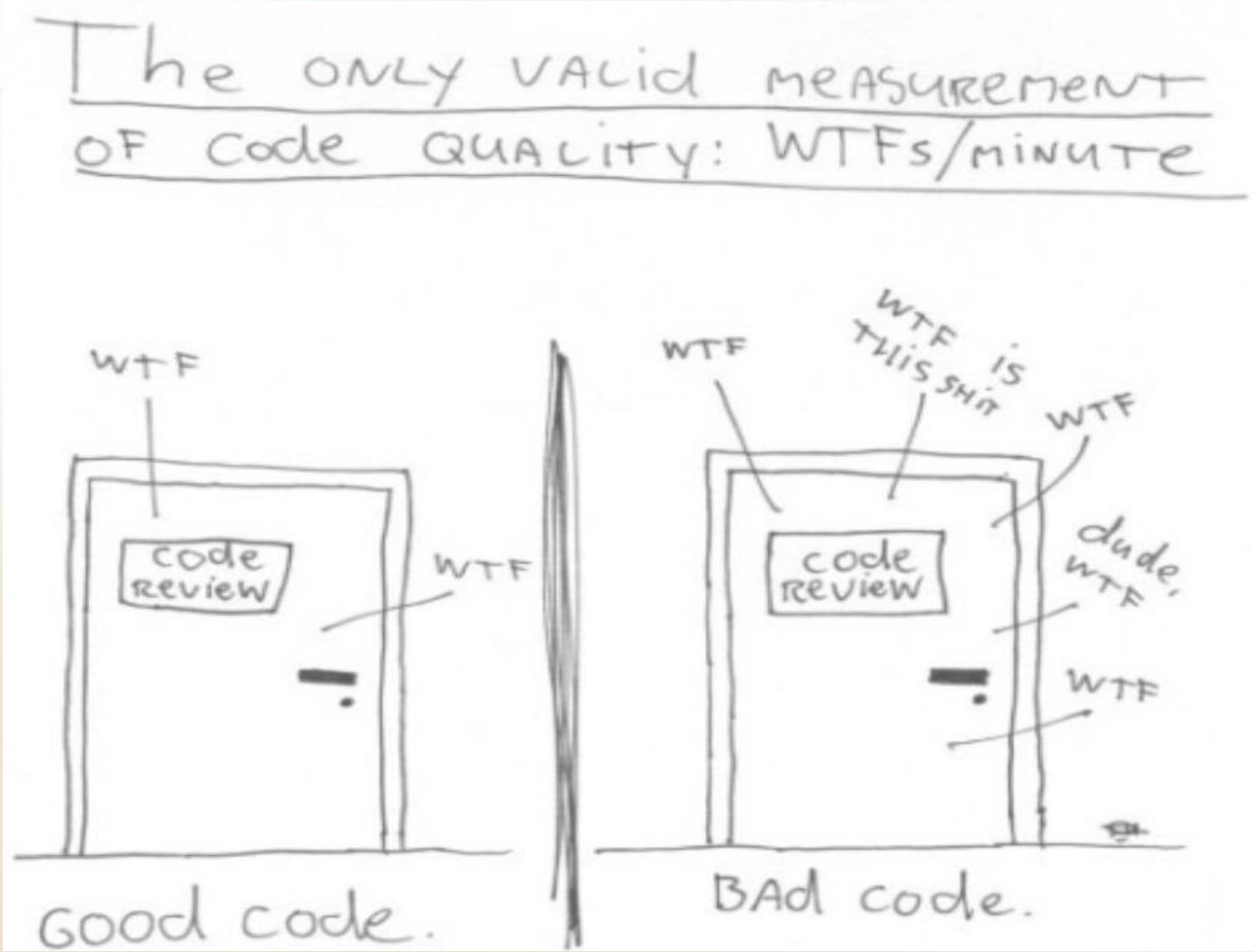
Bra design?

- Synbart när ändringar behöver genomföras
- Synbart när människor läser kod
- Systemdelar är isolerade från varandra
- Grunden är oerhört betydelsefull
- **Testbarhet är nyckeln!**



XLENT
CONSULTING GROUP

Bra design?



XLENT
CONSULTING GROUP

Designmönster

- Gemensamt språk

XLENT
CONSULTING GROUP

Designmönster

- Gemensamt språk

```
class PersonFactory:

    def create_person(self):
        return Person()
```

XLENT
CONSULTING GROUP

Designmönster

- Gemensamt språk
- Är egentligen en lösningsidé som utelämnar detaljerna

XLENT
CONSULTING GROUP

Designmönster

- Gemensamt språk
- Är egentligen en lösningsidé som utelämnar detaljerna
- Underlättar framtida förändringar!

XLENT
CONSULTING GROUP

Designmönster

- Utgå aldrig från ett designmönster vid kodning!
- Låt alltid problemet leda fram till ett designmönster!

XLENT
CONSULTING GROUP

Designmönster

XLENT
CONSULTING GROUP

Problemformuleringar

- *Varför har min kod så svårt att klara en förändring?*
- *Varför har jag skapat en komponent som gör allt?*

XLENT
CONSULTING GROUP

Problemformuleringar

- *Varför har min kod så svårt att klara en förändring?*
  - Hur kan *Dependency Injection* hjälpa?
- *Varför har jag skapat en komponent som gör allt?*
  - Hur kan *Command* hjälpa?

XLENT
CONSULTING GROUP

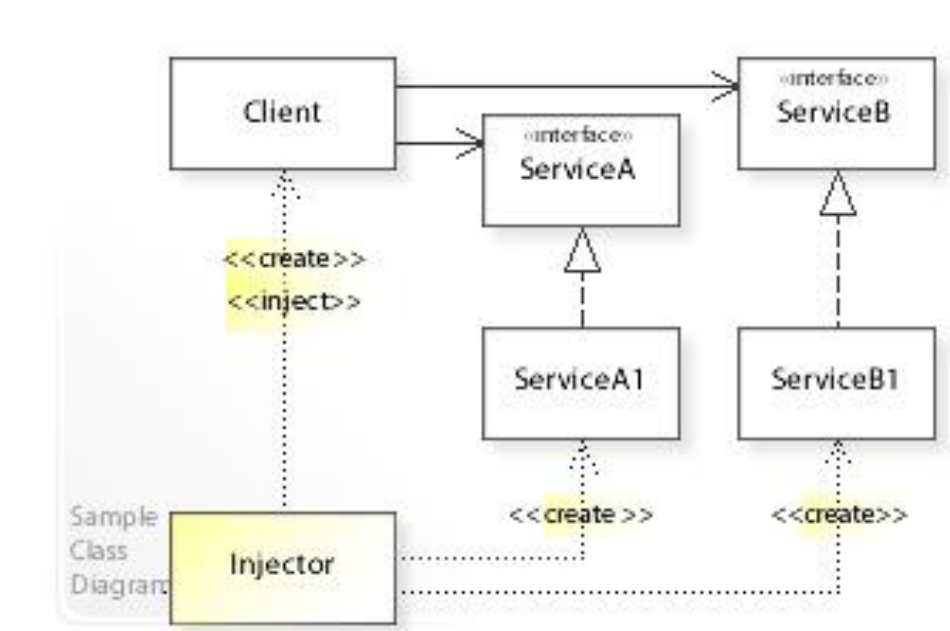
Dependency Injection

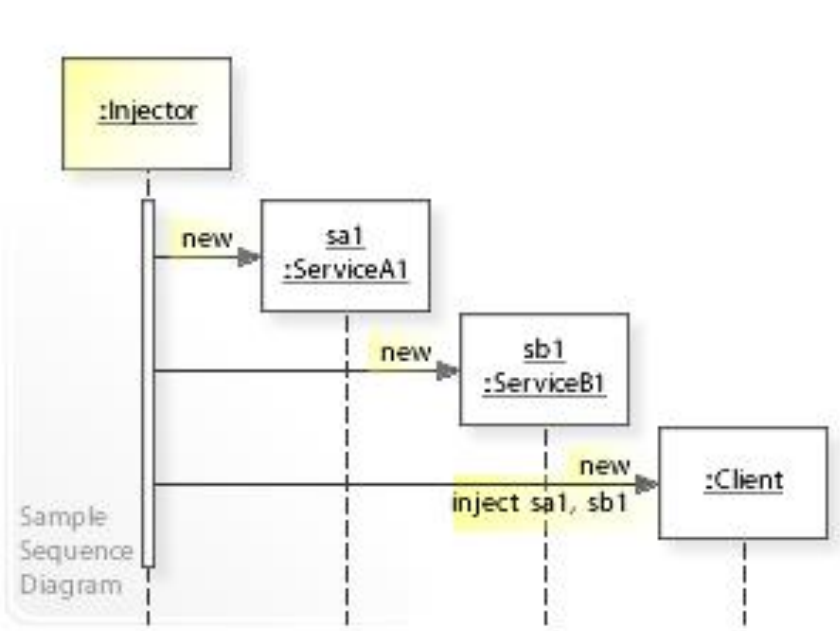
- Innebär att anropande system bestämmer vad som ska användas.

XLENT
CONSULTING GROUP

Dependency Injection

- Innebär att anropande system bestämmer vad som ska användas.





XLENT
CONSULTING GROUP

Dependency Injection

***Utan** Dependency Injection*

```
class OrderService:

    def do_something(self):
        repository = OrderRepository()
        repository.call_database()
```

XLENT CONSULTING GROUP

Dependency Injection

- Använd konstruktor och skicka in beroende.

***Utan** Dependency Injection*

```
class OrderService:

   def do_something(self):
      repository = OrderRepository()
      repository.call_database()
```

***Med** Dependency Injection*

```
class OrderService:

   def __init__(self, repository):
      self.repository = repository


   def do_something(self):
      repository.call_database()
```

XLENT
CONSULTING GROUP

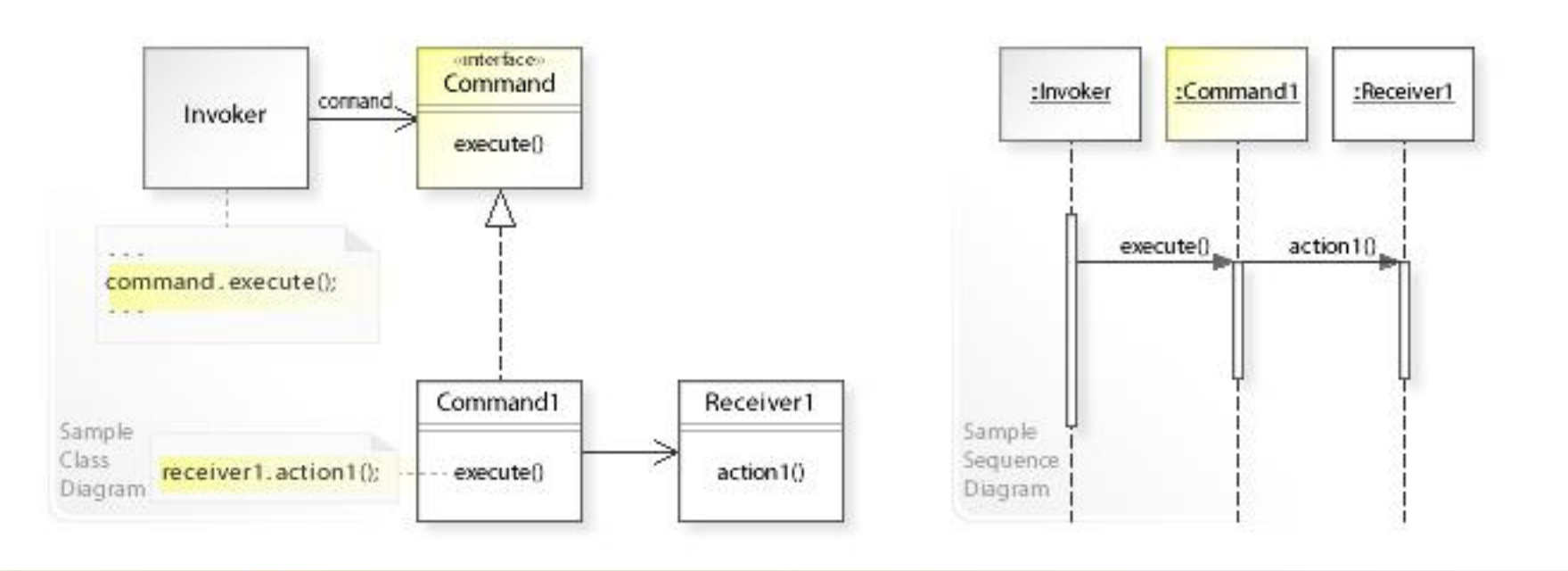
Command

- Command innebär att en operation utförs av en enskild komponent.
- Det är en handling, något som händer i systemet.
- Den enskilda komponenten har all information som behövs för operationen.

XLENT
CONSULTING GROUP

Command

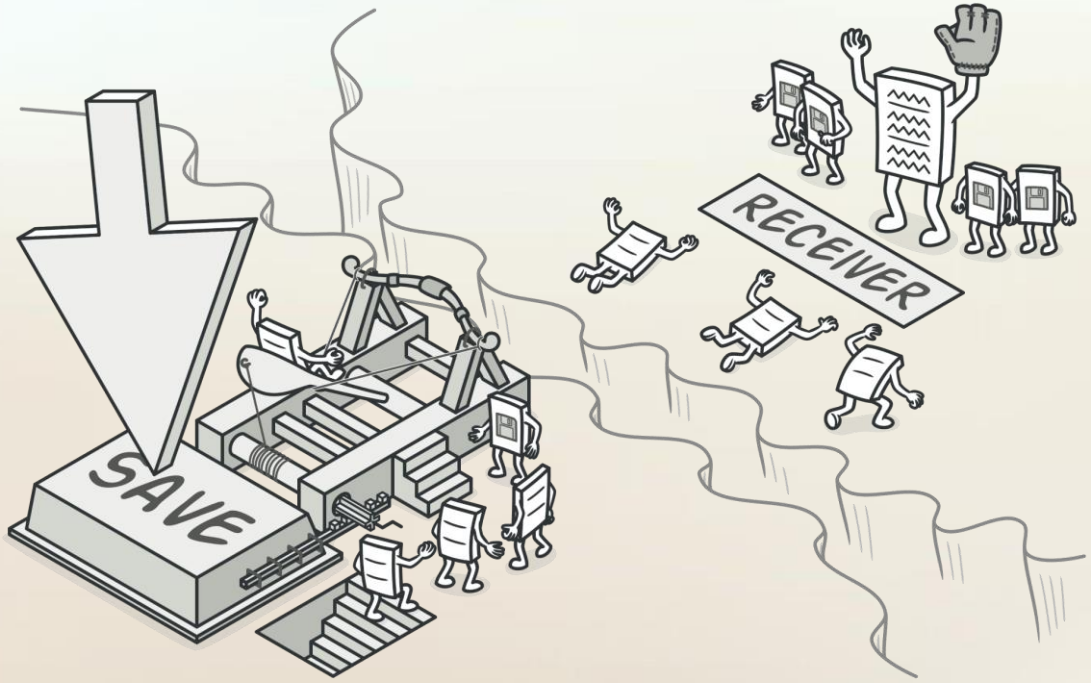
- Command innebär att en operation utförs av en enskild komponent.
- Det är en handling, något som händer i systemet.
- Den enskilda komponenten har all information som behövs för operationen.



XLENT
CONSULTING GROUP

Command

- Command innebär att en operation utförs av en enskild komponent.
- Det är en handling, något som händer i systemet.
- Den enskilda komponenten har all information som behövs för operationen.



XLENT
CONSULTING GROUP

Litteratur:

- *Dependency Injection Principles, Practices, and Patterns* – Steven van Deursen and Mark Seemann (2019)
- *Clean Code* – Robert C Martin (2008)
- Clean Architecture – Robert C Martin (2017)
- *The Art of Unit Testing* – Roy Osherove (2014)
- *Pluralsight* – online learning platform

XLENT
CONSULTING GROUP

Presentationen hittas här:

- https://github.com/starefeldt/DesignPatternsInPython