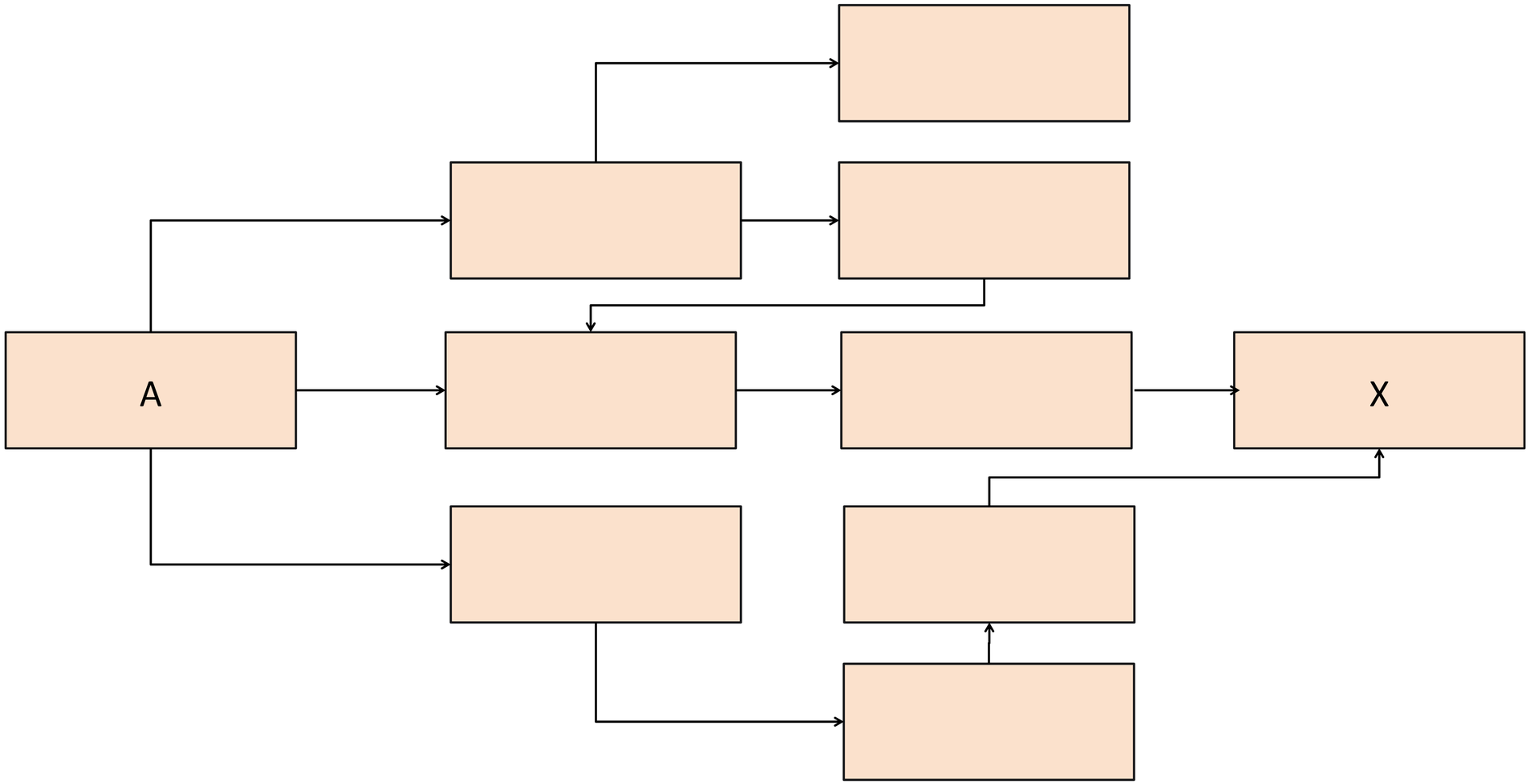
Kopplung

„Beziehungen (=Kopplung) zwischen Bausteinen ermöglichen deren Zusammenarbeit, bilden also die Grundsubstanz kooperierender Strukturen. Andererseits bedeuten Beziehungen auch Abhängigkeiten zwischen Bausteinen.“

Gernot Starke: Effektive Softwarearchitekturen: Ein praktischer Leitfaden. M: Carl Hanser Verlag, 2015.


A
X

Benachrichtigungskopplung

Aufrufkopplung

Datenkopplung

Kontrollkopplung

Externe Kopplung

Gemeinschaftliche Kopplung

Inhaltskopplung

Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.

Benachrichtigungskopplung

Aufrufkopplung

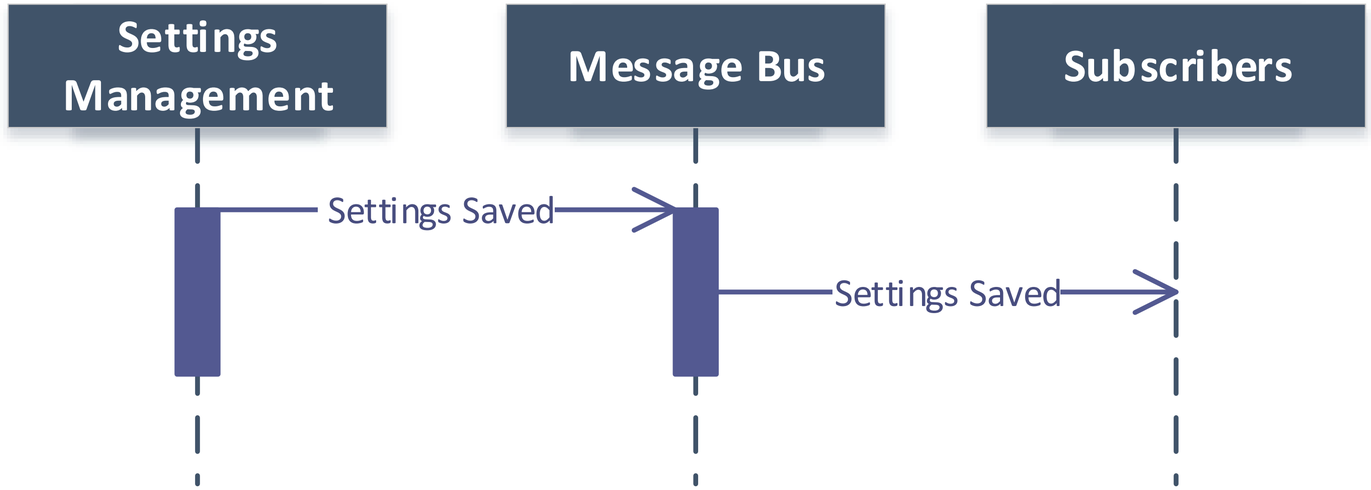
Datenkopplung

Kontrollkopplung

Externe Kopplung

Gemeinschaftliche Kopplung

Inhaltskopplung



Benachrichtigungskopplung

Aufrufkopplung

Datenkopplung

Kontrollkopplung

settings.Save()

Externe Kopplung

Gemeinschaftliche Kopplung

Inhaltskopplung

Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.

Benachrichtigungskopplung

Aufrufkopplung

Datenkopplung

```
settings.GetValue("Max")
```

Kontrollkopplung

```
var setting = new Setting("Max", 6)
Settings.Save(setting)
```

Externe Kopplung

Gemeinschaftliche Kopplung

Inhaltskopplung

Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.

Benachrichtigungskopplung

Aufrufkopplung

Datenkopplung

Kontrollkopplung

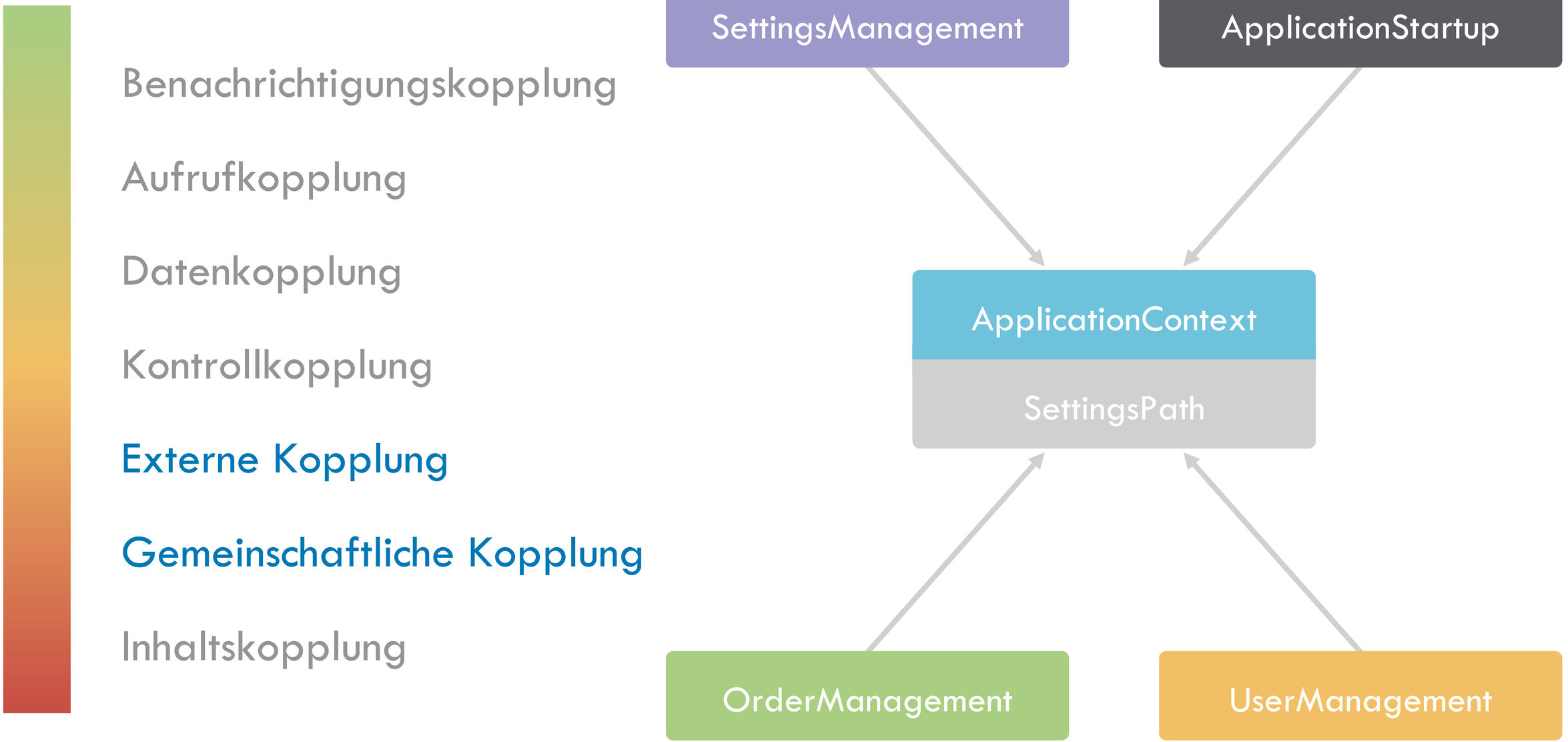
Externe Kopplung

Gemeinschaftliche Kopplung

Inhaltskopplung

```
var setting = new Setting()
...
settings.Update(setting, true)
```

Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.



Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.

Benachrichtigungskopplung

Aufrufkopplung

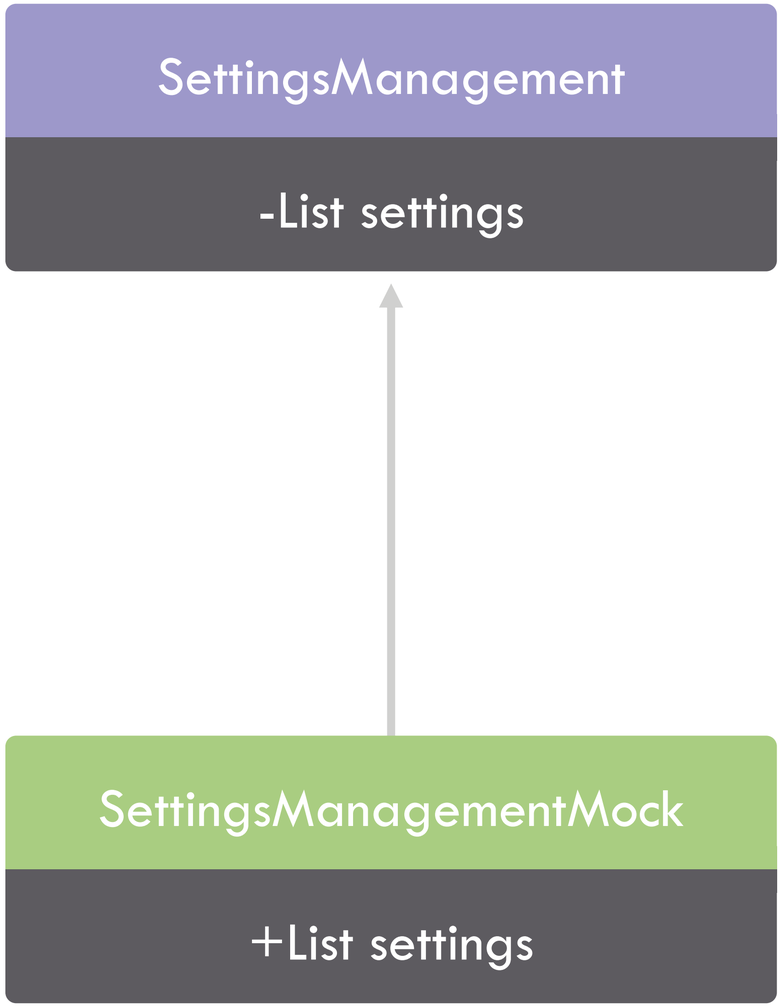
Datenkopplung

Kontrollkopplung

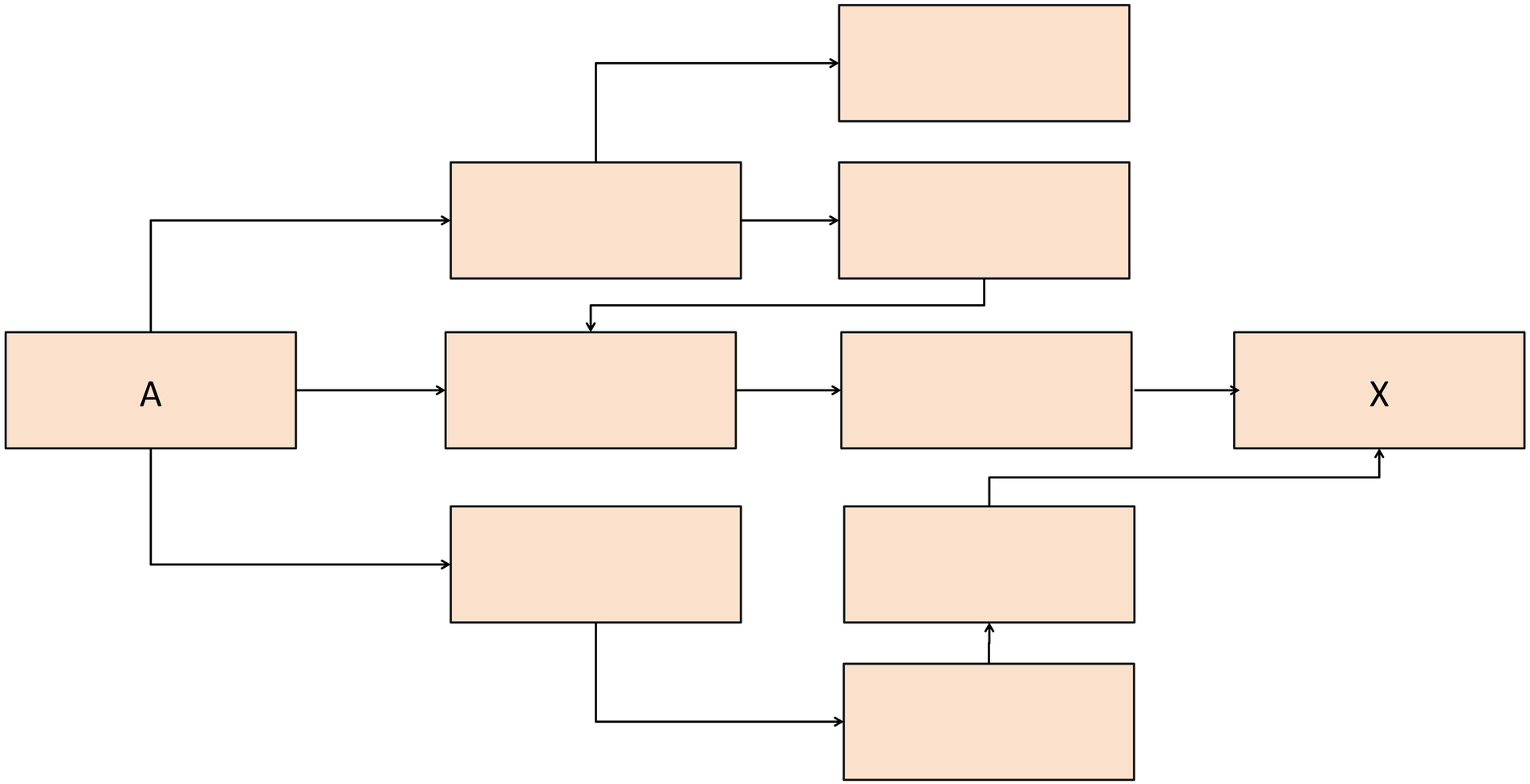
Externe Kopplung

Gemeinschaftliche Kopplung

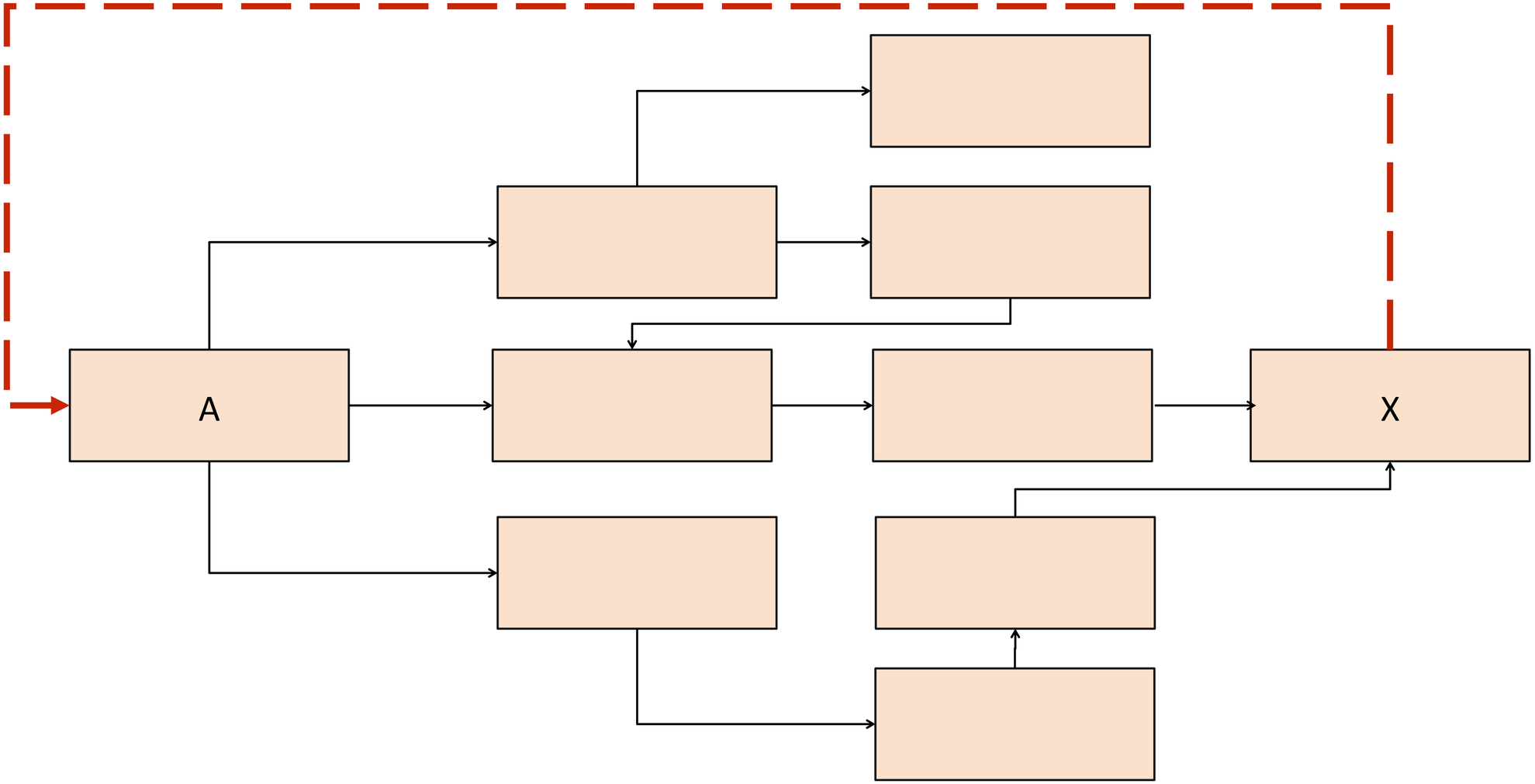
Inhaltskopplung

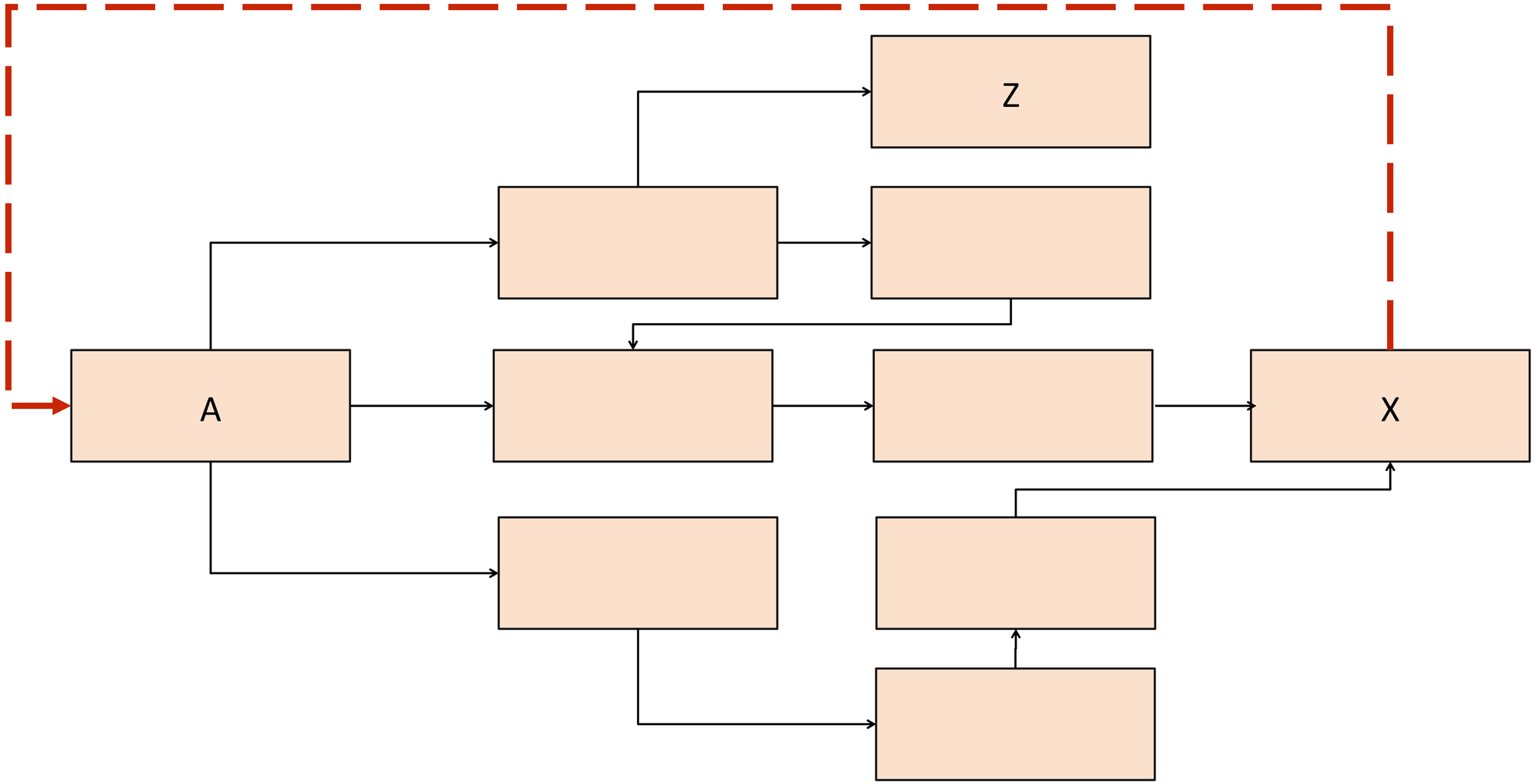


Basierend auf Glenford J. Myers: *Reliable Software through Composite Design.* Mason and Lipscomb Publishers, New York 1974.


A
X

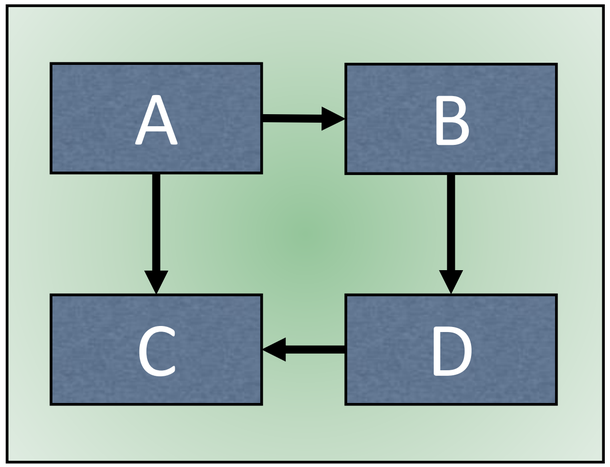
- Rot gestrichelt wichtig

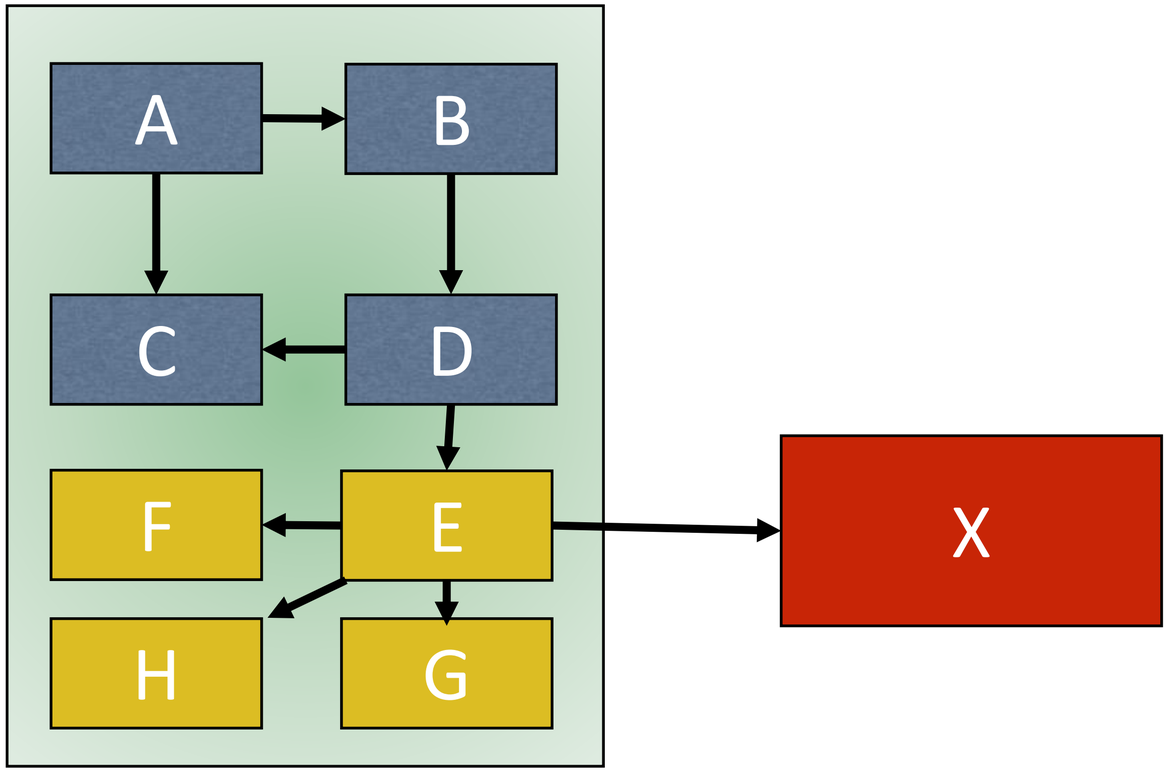


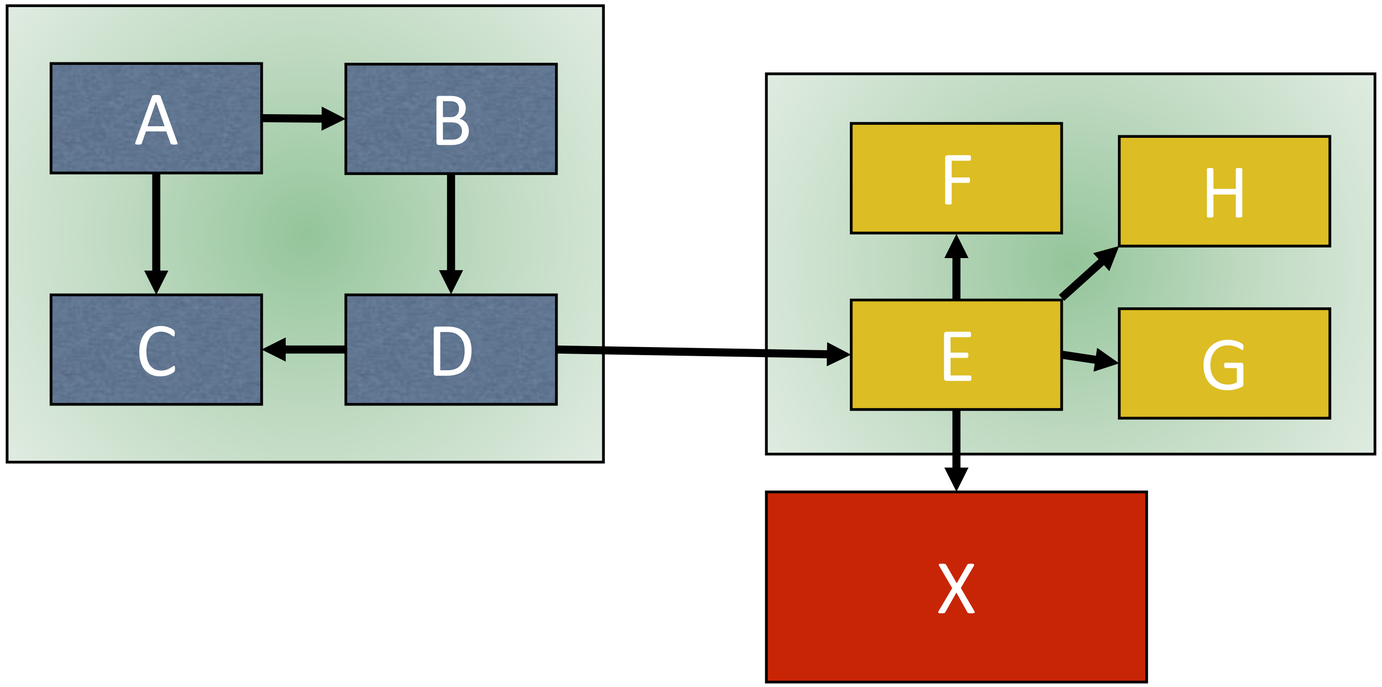

Z
A
X

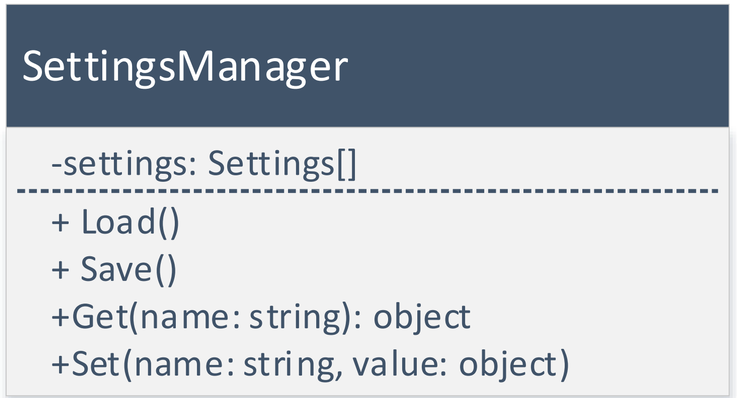
Kohäsion

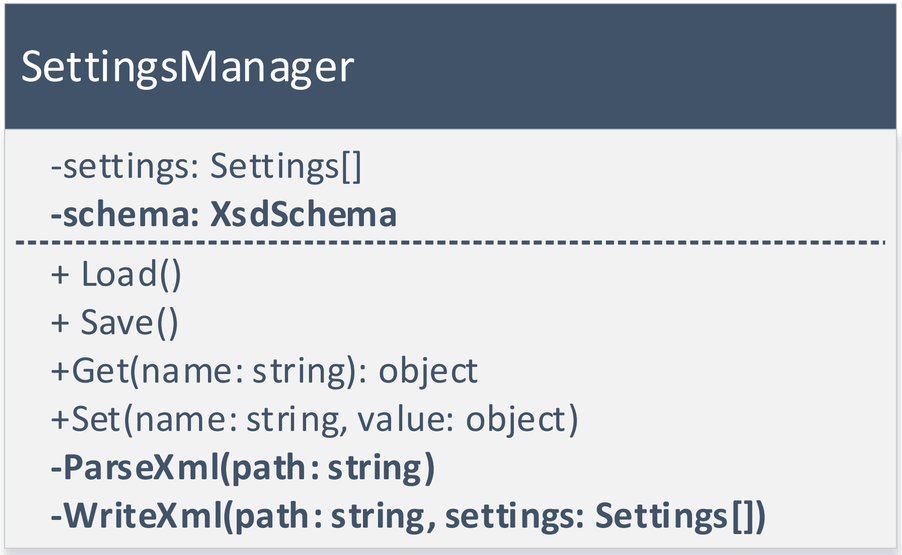
Kohäsion = Bindung = Innerer Zusammenhang

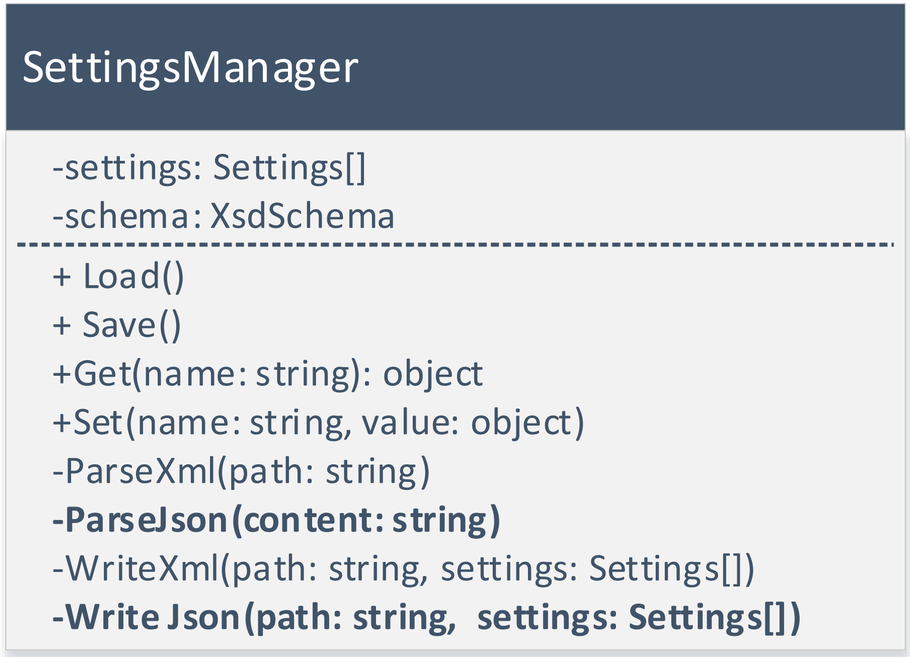

A
B
C
D

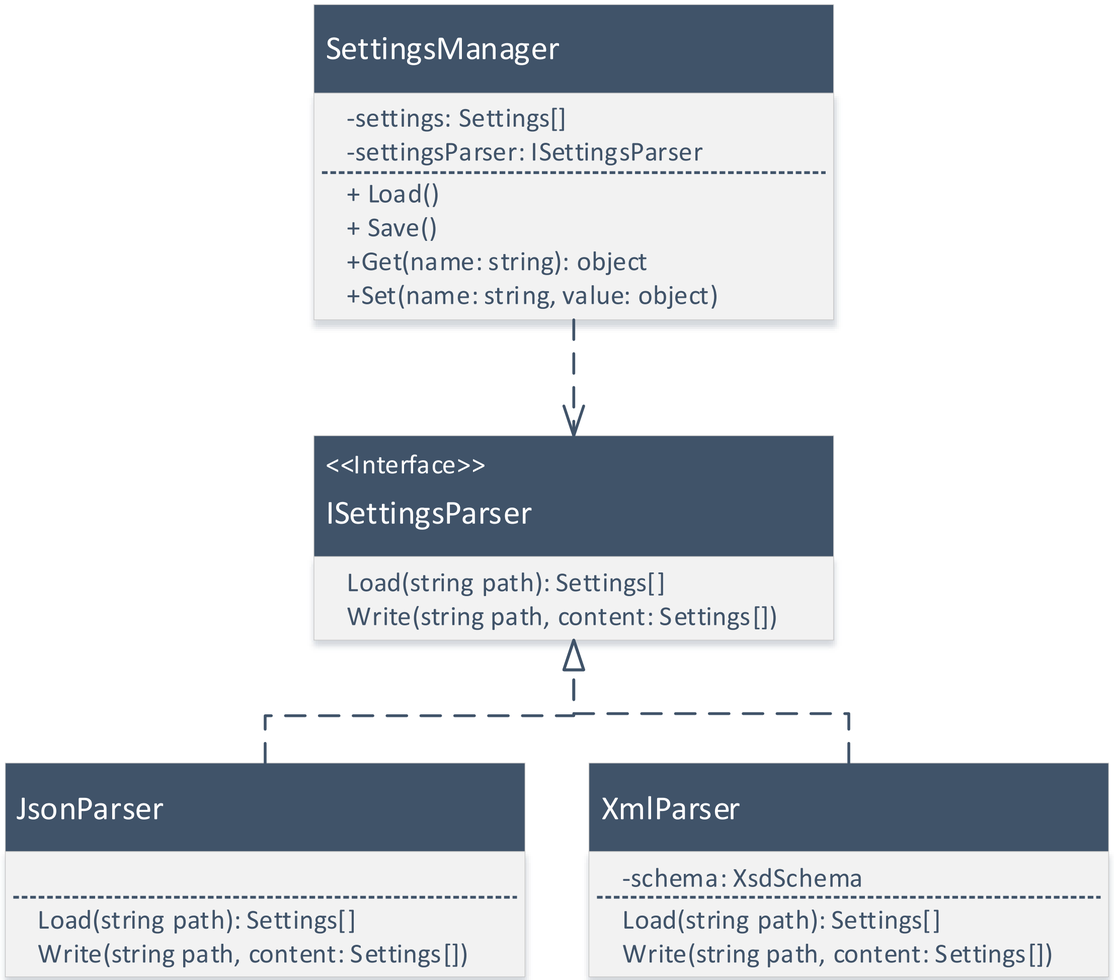

A
B
C
D
F
E
H
G
X


A
B
C
D
F
H
E
G
X


SettingsManager
-settings: Settings[]
+ Load()
+ Save()
+Get(name: string): object
+Set(name: string, value: object)


SettingsManager
-settings: Settings[]
-schema: XsdSchema
+ Load()
+ Save()
+Get(name: string): object
+Set(name: string, value: object)
-ParseXml(path: string)
-WriteXml(path: string, settings: Settings[])


SettingsManager
-settings: Settings[]
-schema: XsdSchema
+ Load()
+ Save()
+Get(name: string): object
+Set(name: string, value: object)
-ParseXml(path: string)
-ParseJson(content: string)
-WriteXml(path: string, settings: Settings[])
-Write Json(path: string, settings: Settings[])


SettingsManager
-settings: Settings[]
-settingsParser: ISettingsParser
+ Load()
+ Save()
+Get(name: string): object
+Set(name: string, value: object)
<<Interface>>
ISettingsParser
Load(string path): Settings[]
Write(string path, content: Settings[])
JsonParser
Load(string path): Settings[]
Write(string path, content: Settings[])
XmlParser
-schema: XsdSchema
Load(string path): Settings[]
Write(string path, content: Settings[])

Ziele

- Enge innere Bindung
- Lose Kopplung

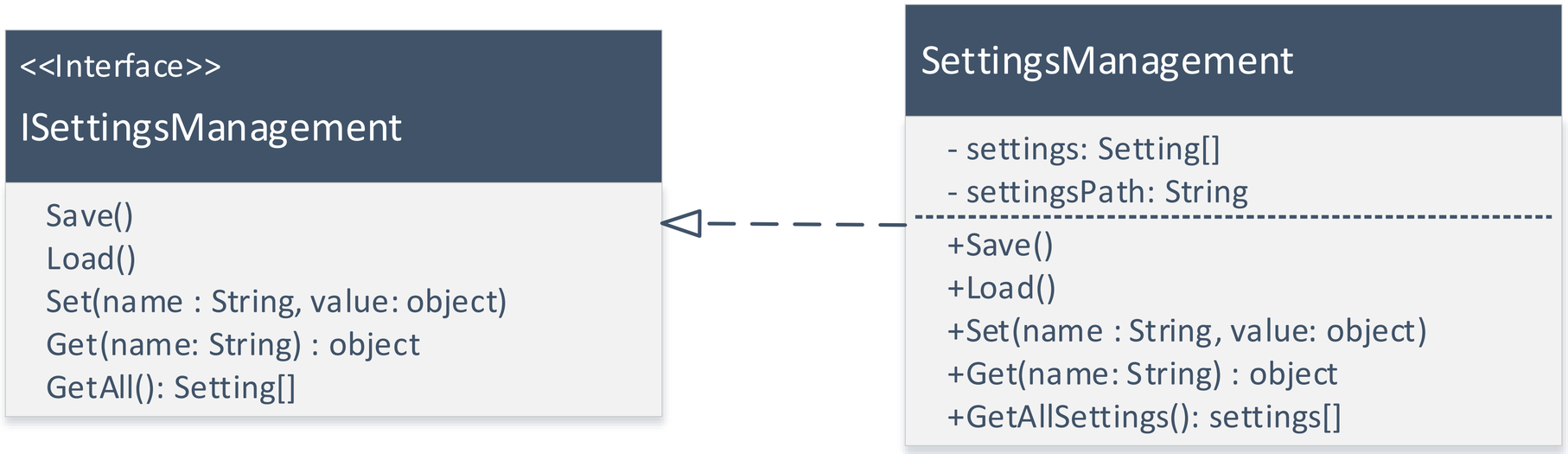
Entkopplung & Abstraktion

Lose Kopplung

Abstraktion

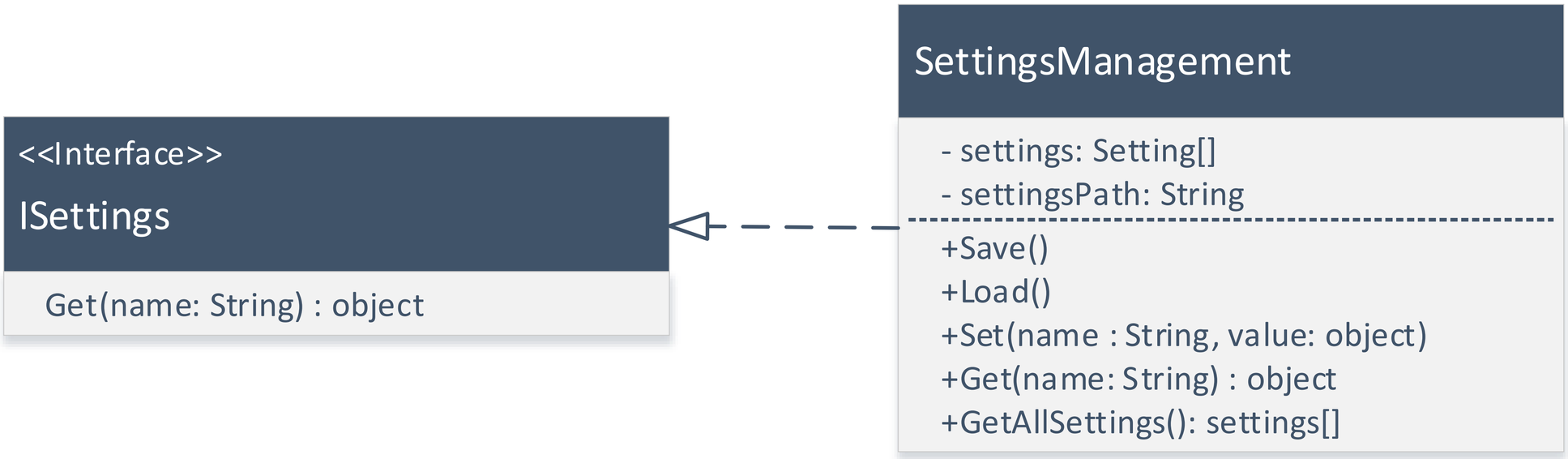
was

wie



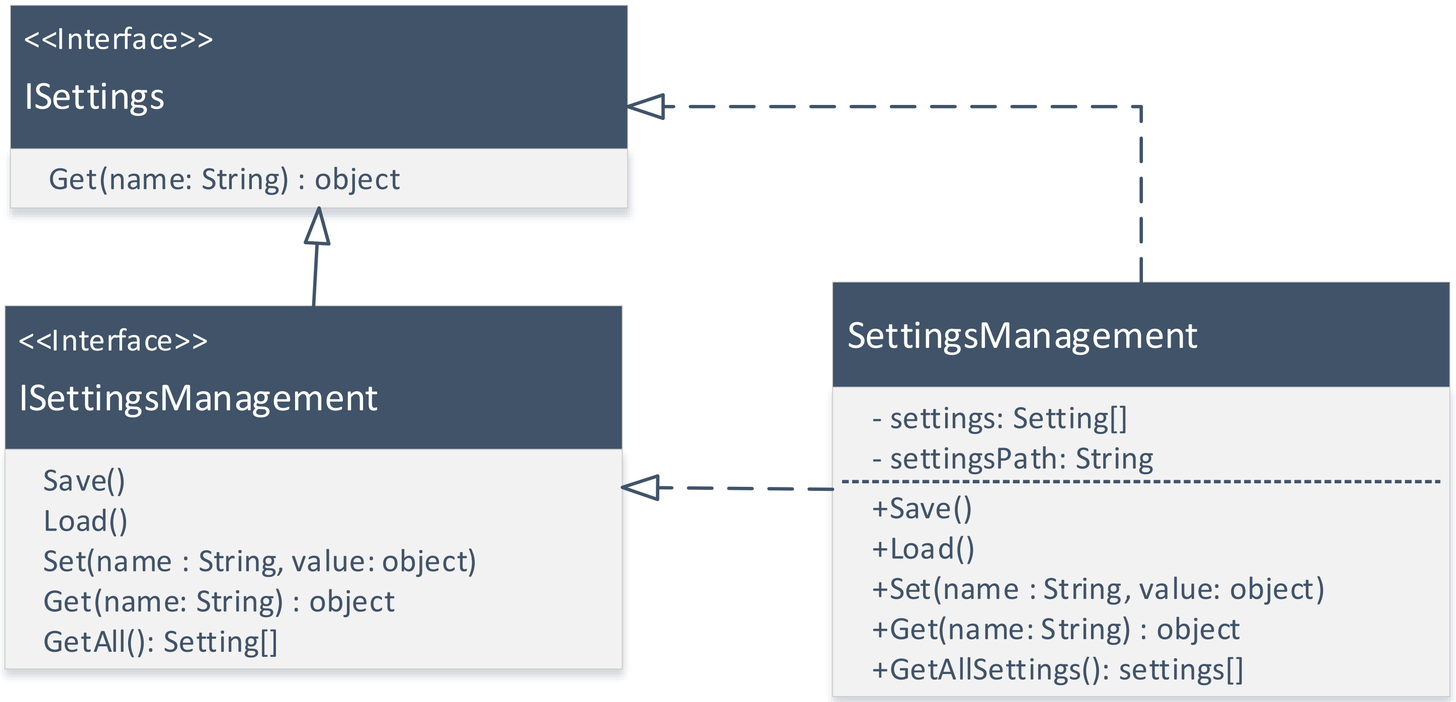
was

wie


<<Interface>>
ISettings
Get(name: String) : object
SettingsManagement
- settings: Setting[]
- settingsPath: String
+Save()
+Load()
+Set(name : String, value: object)
+Get(name: String) : object
+GetAllSettings(): settings[]

was

wie


<<Interface>>
ISettings
Get(name: String) : object
<<Interface>>
ISettingsManagement
Save()
Load()
Set(name : String, value: object)
Get(name: String) : object
GetAll(): Setting[]
SettingsManagement
- settings: Setting[]
- settingsPath: String
+Save()
+Load()
+Set(name : String, value: object)
+Get(name: String) : object
+GetAllSettings(): settings[]


<<Interface>>
ISettings
Get(name: String) : object
InvoiceService
-settings: ISettings
+Calculate(invoice: Invoice)
<<Interface>>
ISettingsManagement
Save()
Load()
Set(name : String, value: object)
Get(name: String) : object
GetAll(): Setting[]
SettingsManagementViewModel
+Settings : Setting[]
-settingsManagement : ISettingsManagement
Save()
Load()

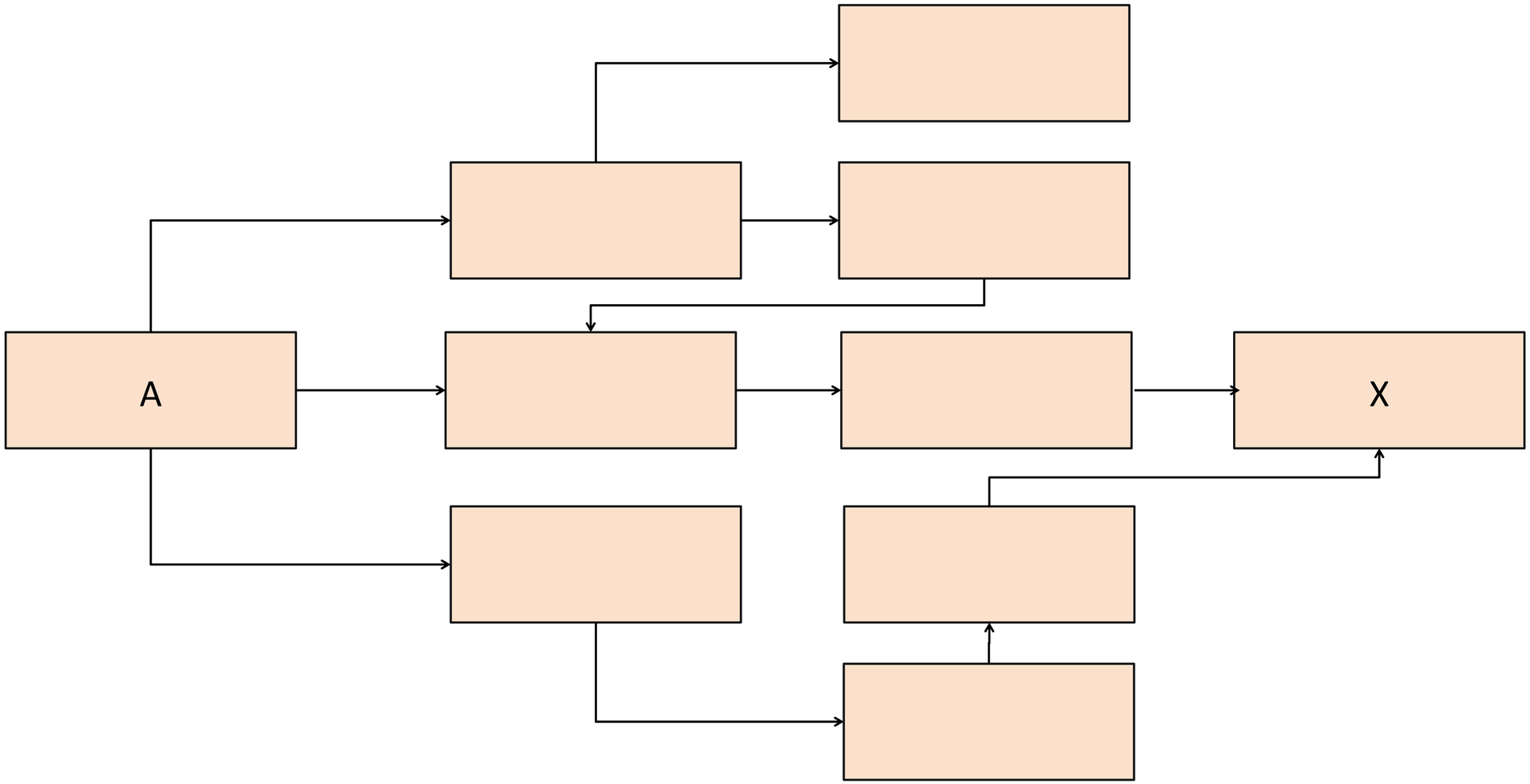
Frühe Bindung

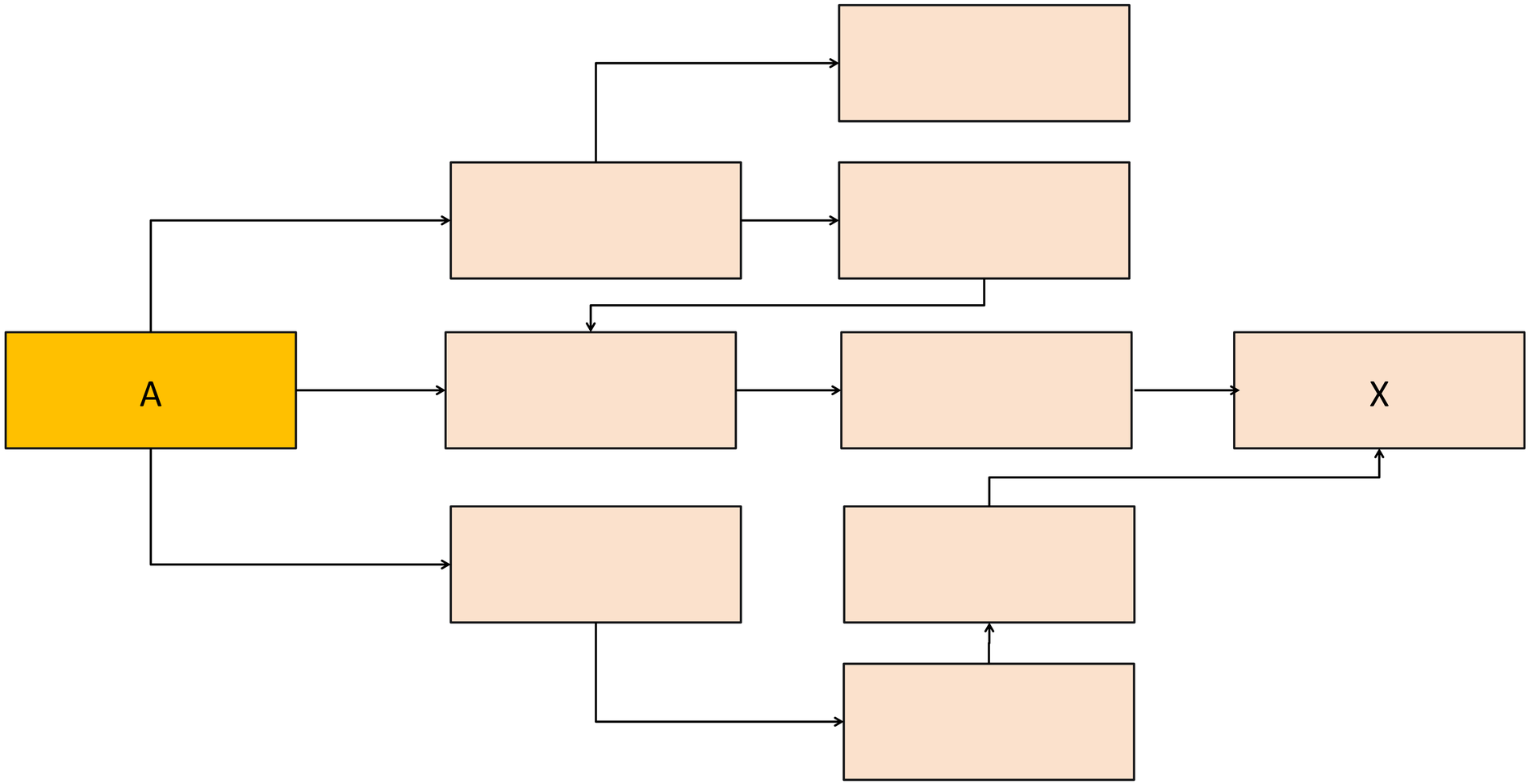
```
public class InvoiceService
{
    private ISettings settings;

     public InvoiceService()
     {
         this.settings = new SettingsManagement()
     }
}
```

Späte Bindung

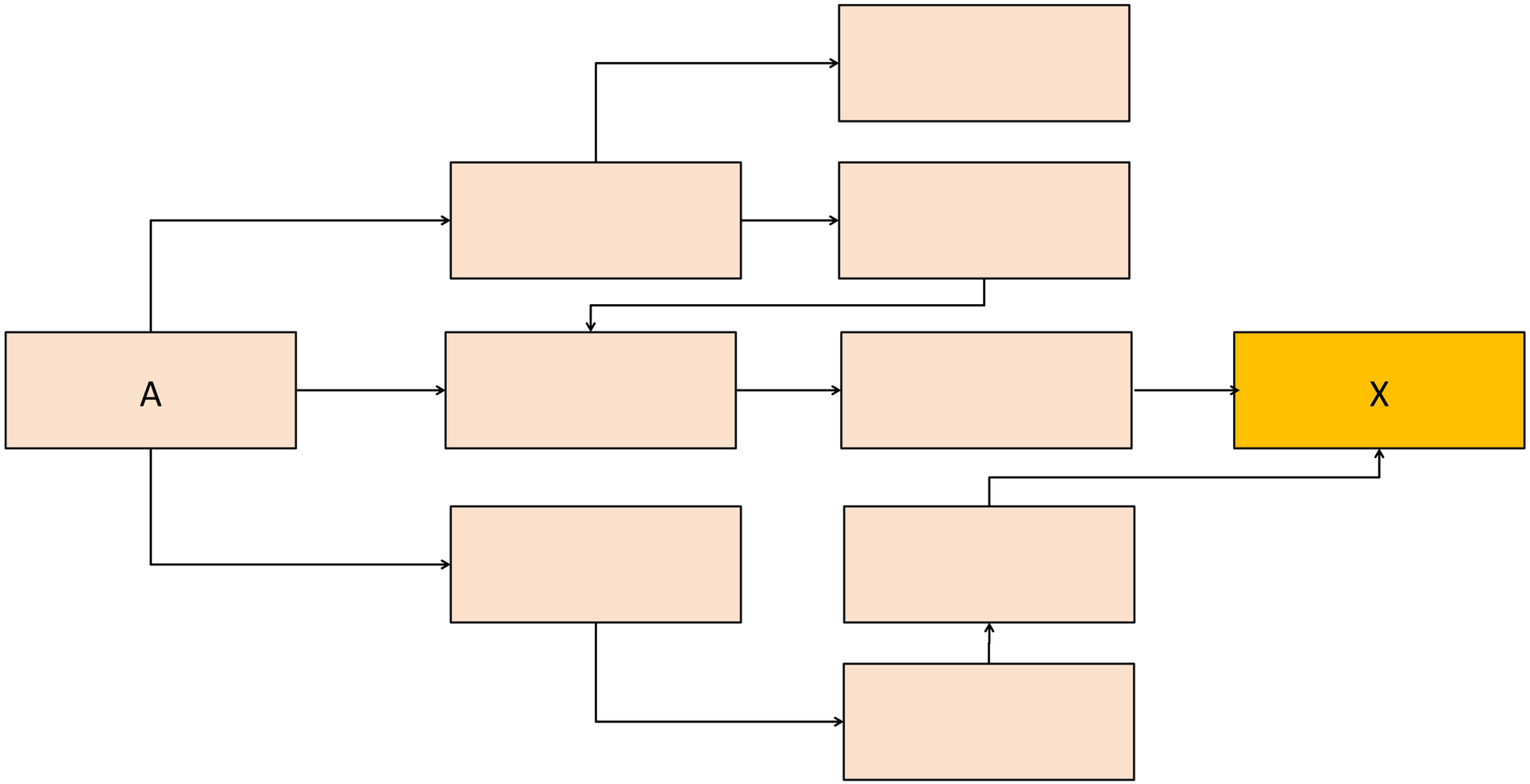
```
public class InvoiceService
{
    private ISettings settings;

     public InvoiceService(ISettings settings)
     {
         this.settings = settings;
     }
}
```

Instabilität vs. Abstraktion


A
X



Efferent Coupling (Ce)



Afferent Coupling (Ca)

$$Instability = Ce / (Ce + Ca)$$

Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

$$Instability = Ce / (Ce + Ca)$$

$$Instability = 2 / (2 + 2) = 0,50$$

$$Instability = 2 / (2 + 4) = 0,33$$

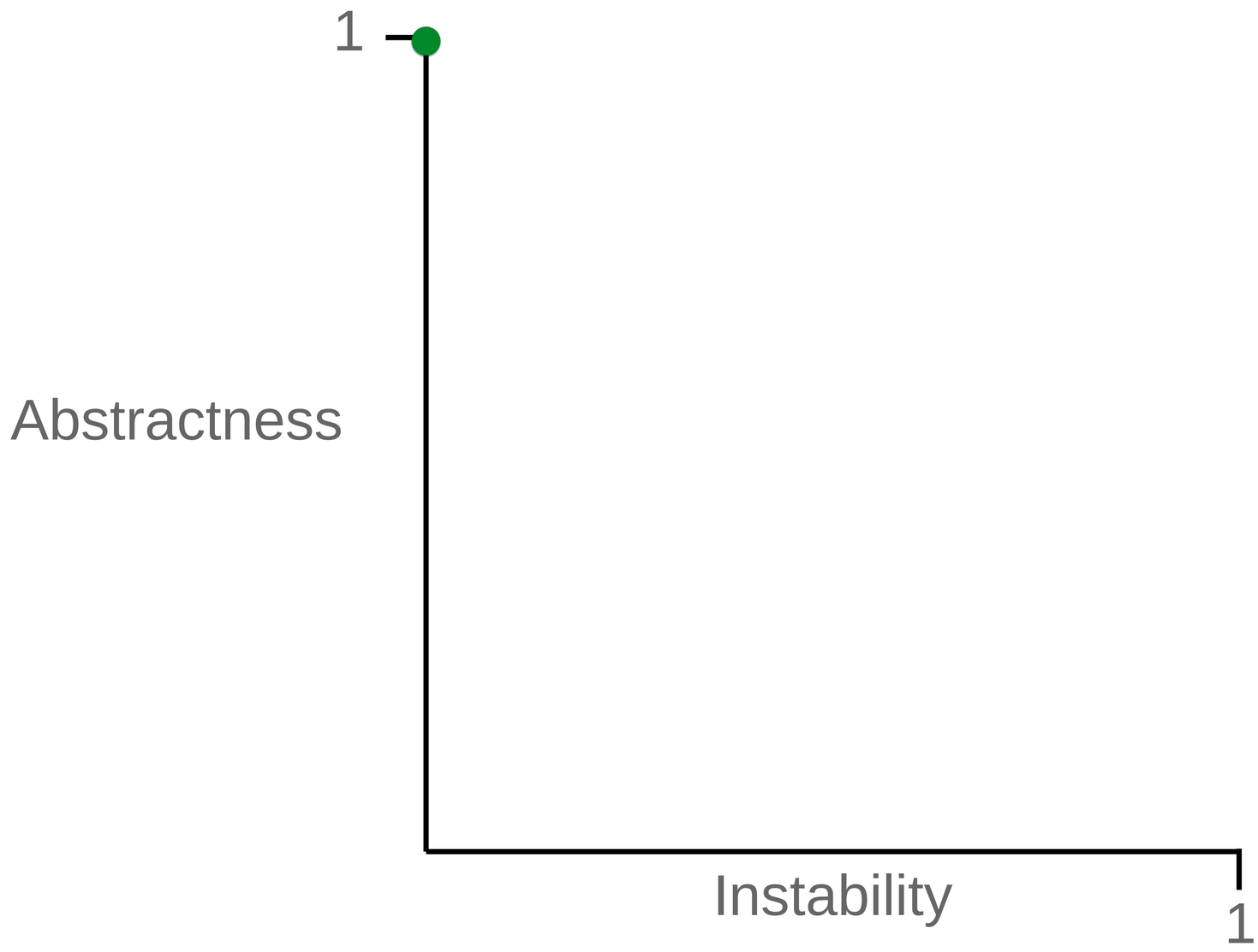
$$Instability = 3 / (3 + 0) = 1,00$$

Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

Abstractness = Na / N

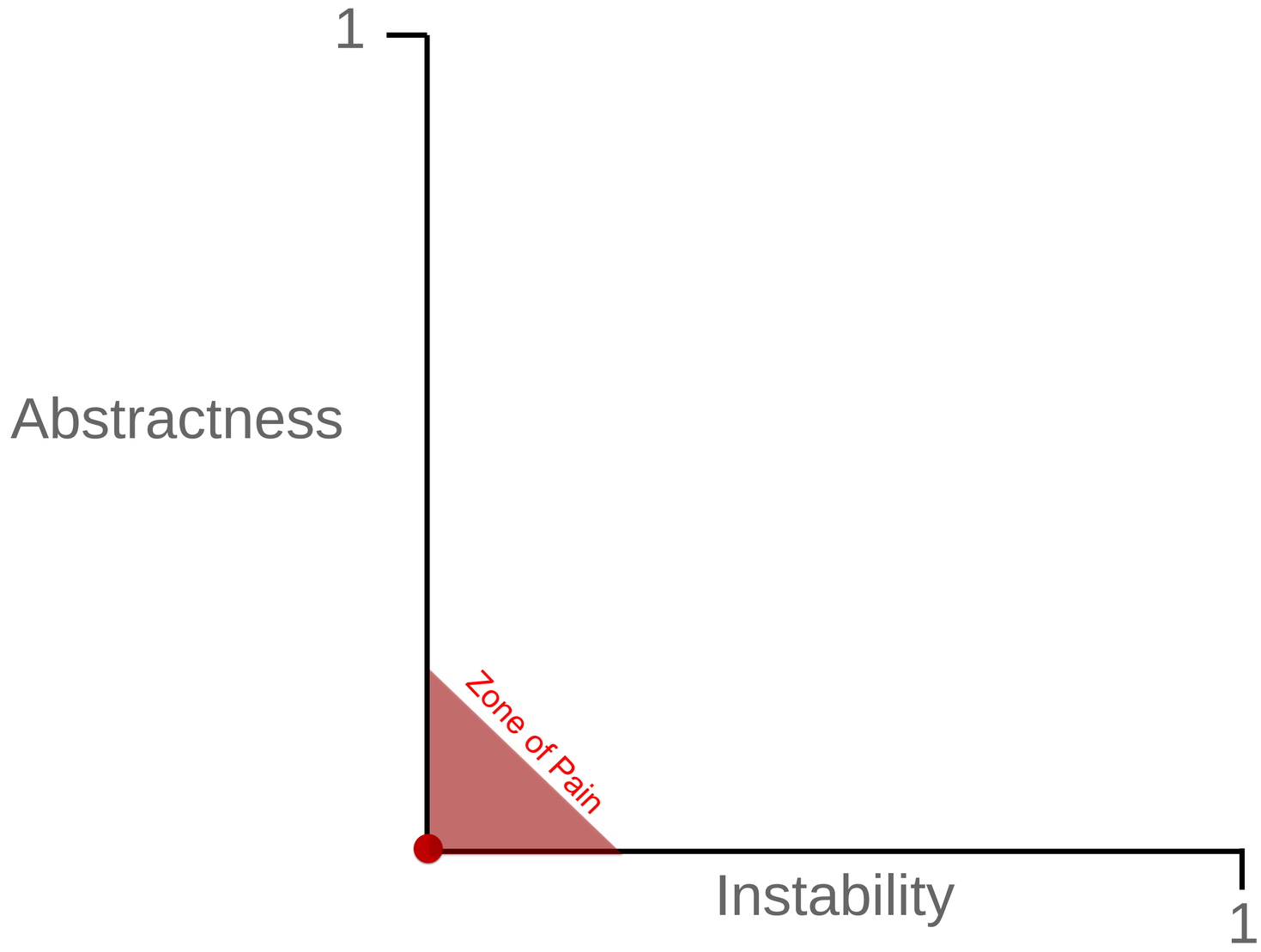
Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

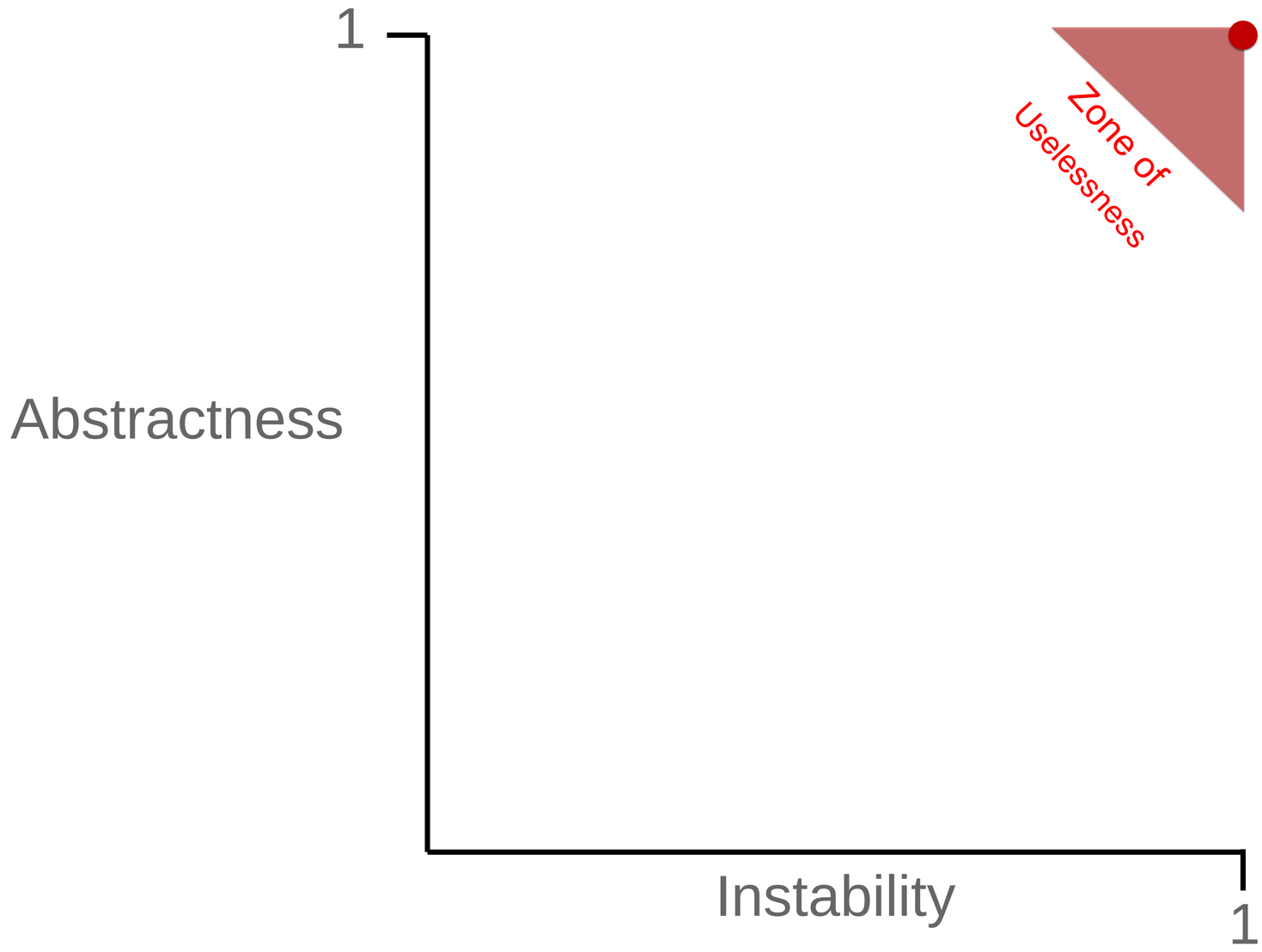


Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

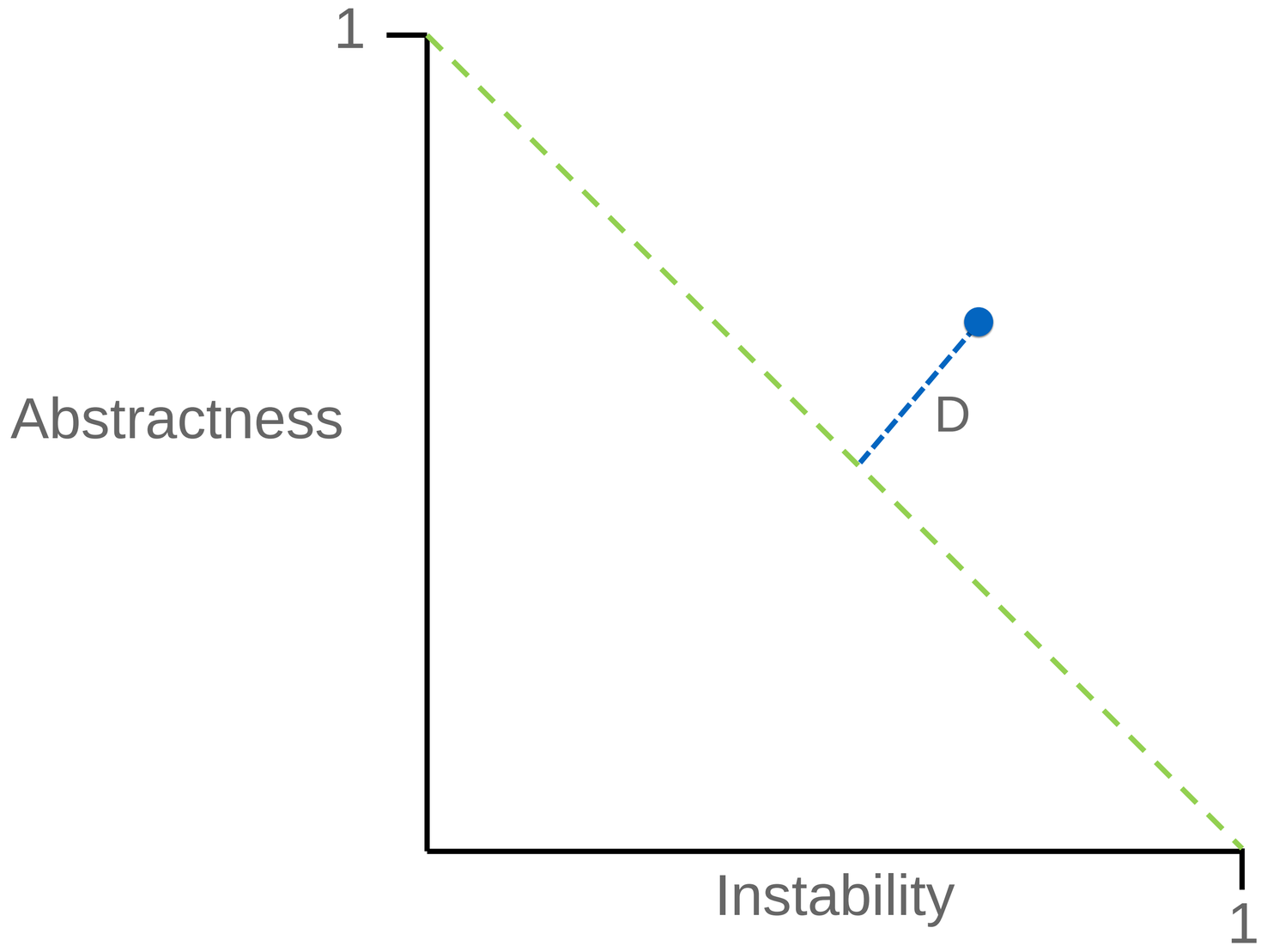
Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod



Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod



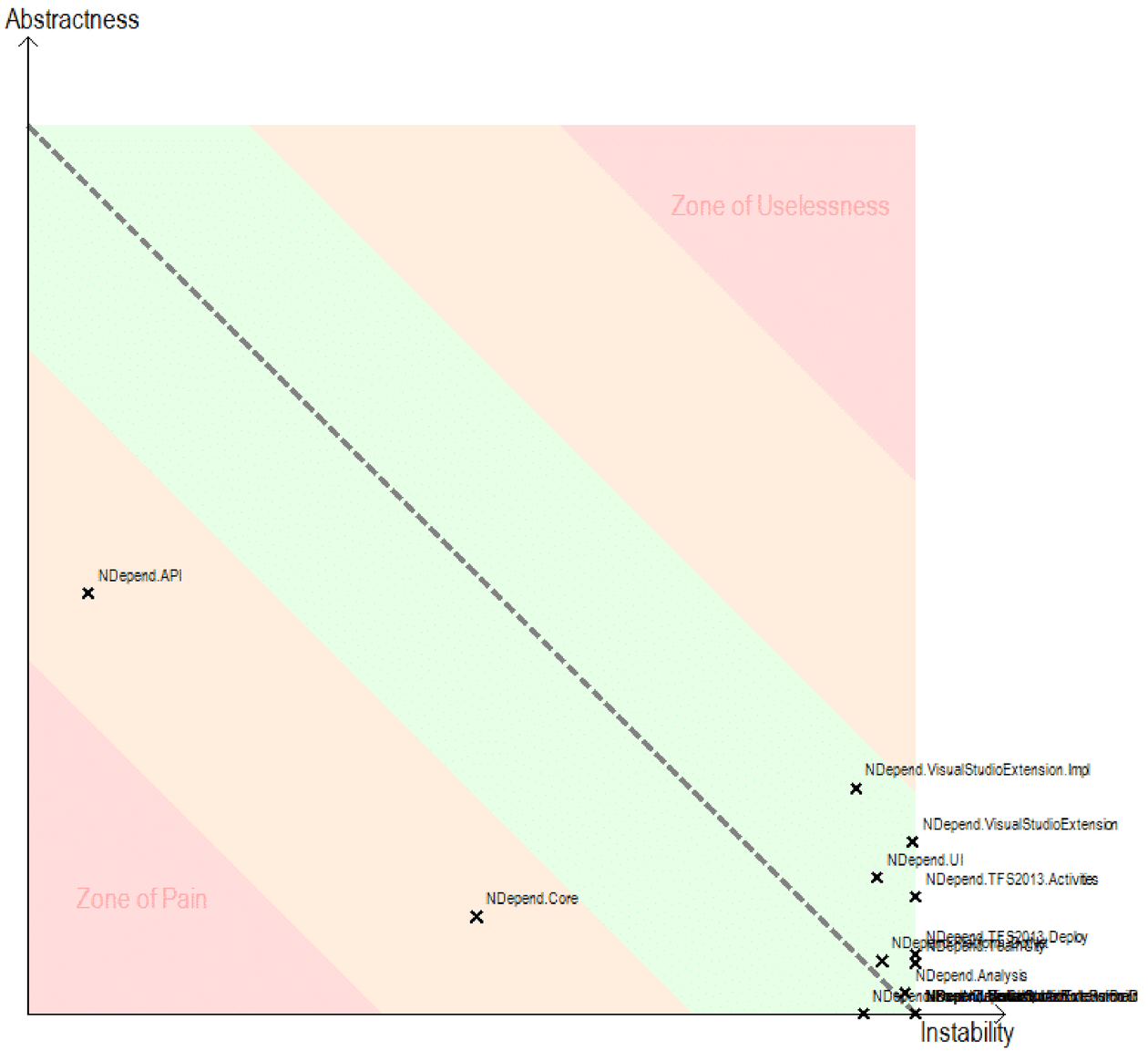
Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod



$$D = \frac{|A + I - 1|}{\sqrt{2}}$$

$$D' = |A + I - 1|$$

Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod



Quelle: https://blog.ndepend.com