



Saxonia Systems

So geht Software.

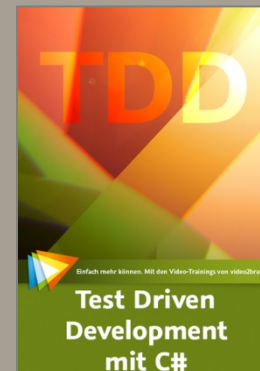
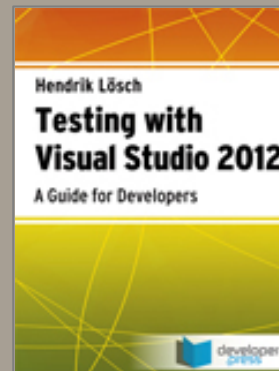


Hendrik Lösch

Hendrik.Loesch@saxsys.de

just-about.net

@HerrLoesch





Saxonia Systems

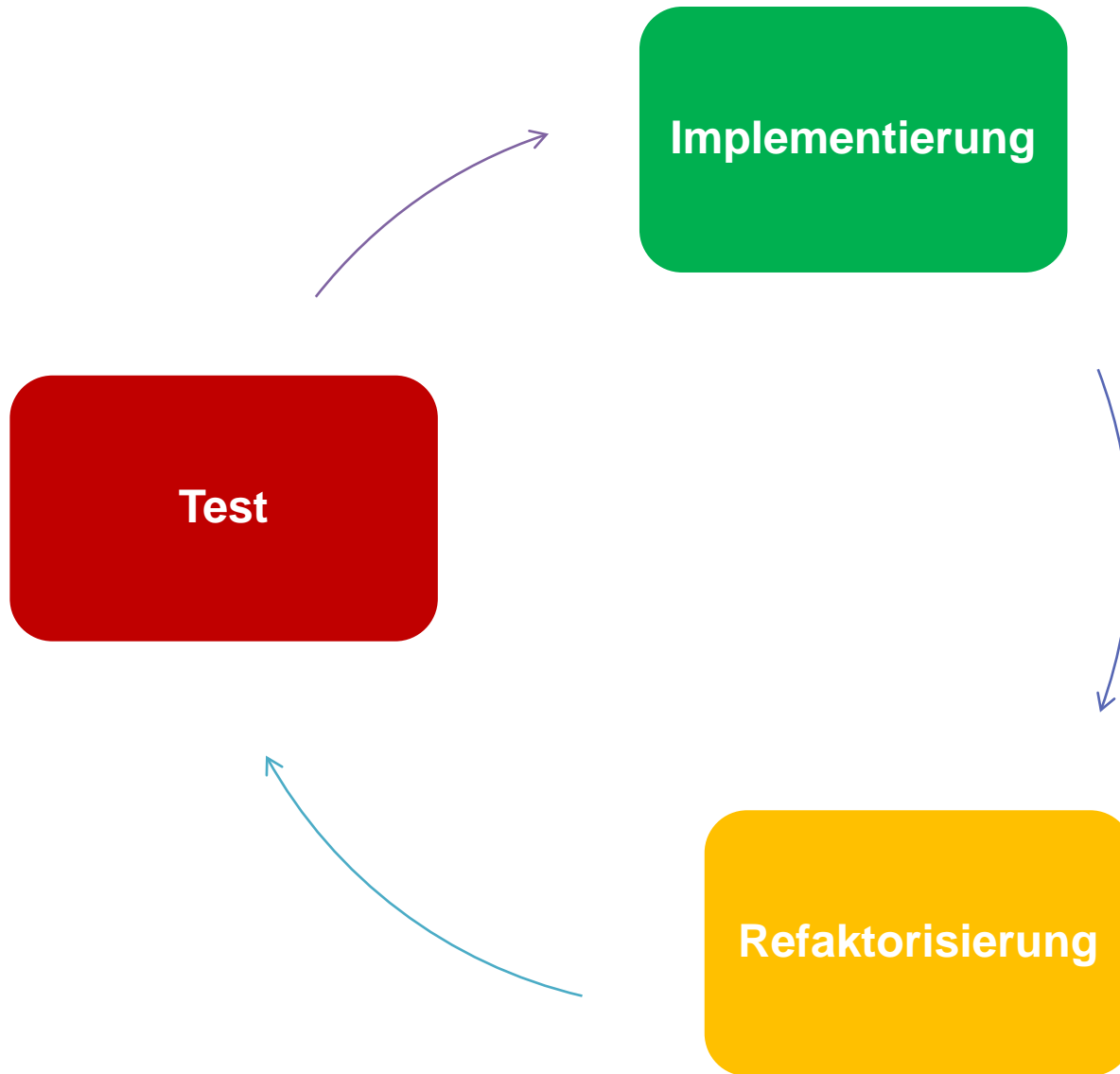
So geht Software.

Test Driven Development

Mysterium und Wagnis?!?

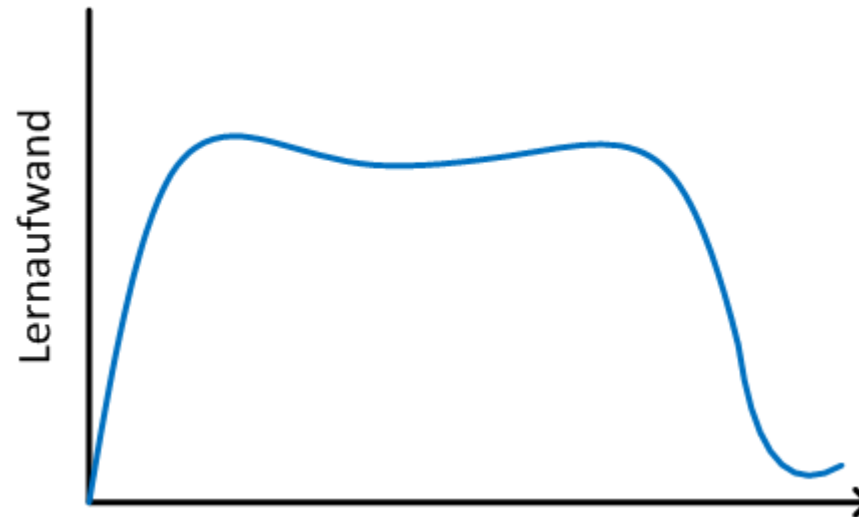
„Wer als einziges Werkzeug einen **Hammer**
hat, sieht in jedem Problem einen **Nagel.**“

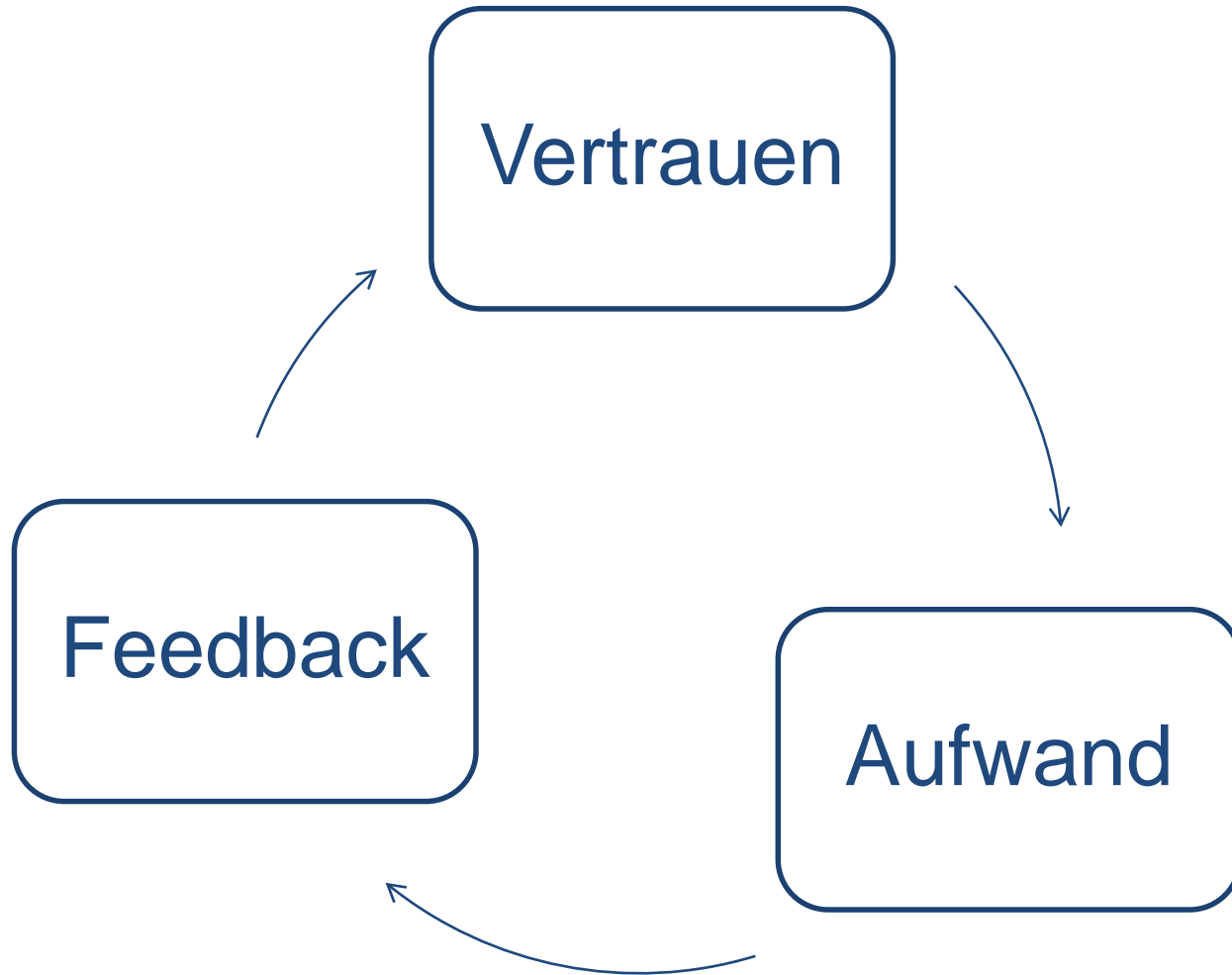
Paul Watzlawick



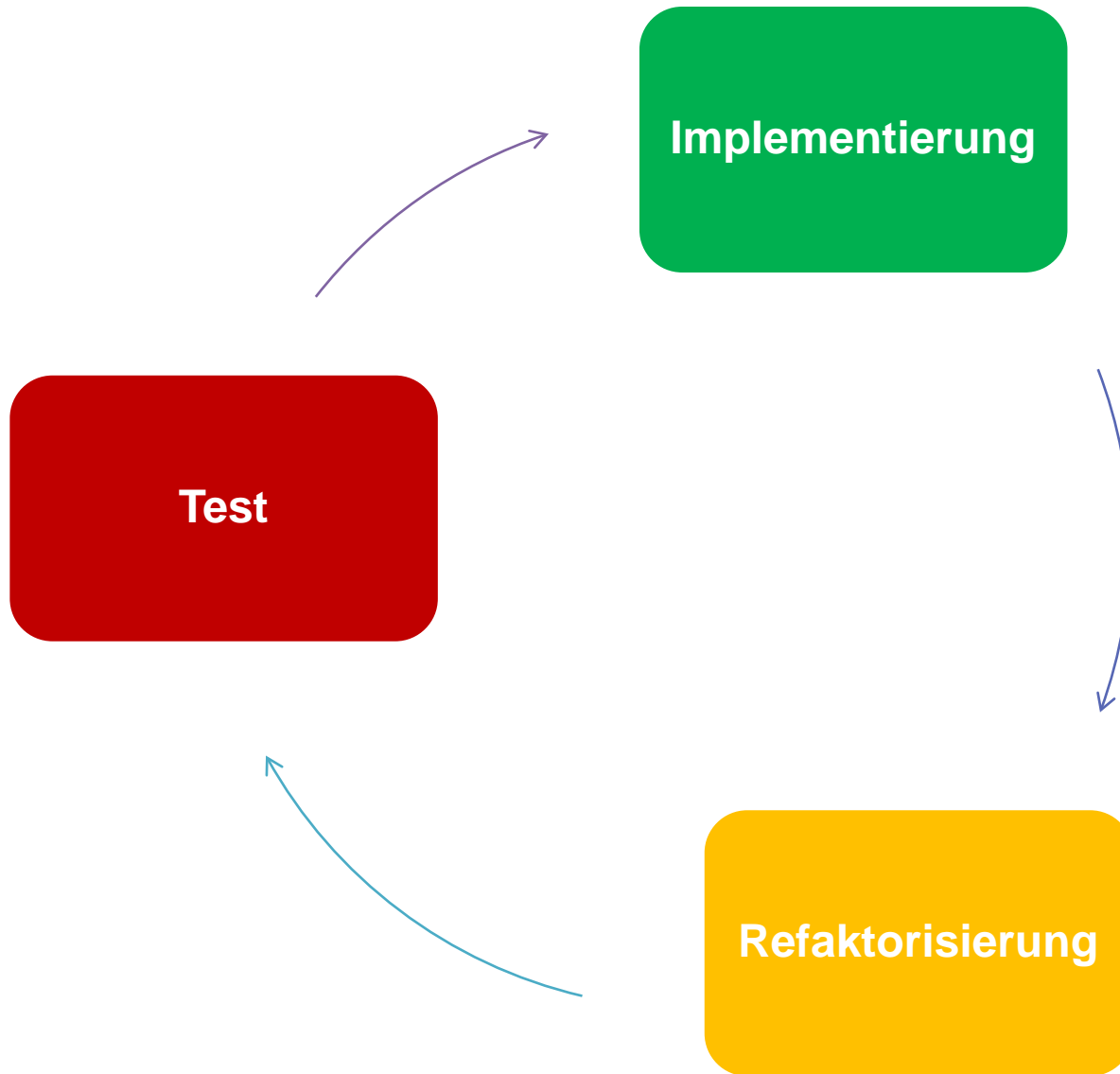
Hump of Pain

“New teams are often expected to adopt practices such as TDD and refactoring, which are difficult to learn. Without good coaching, plenty of time to master new skills, and string management support, they're easily discouraged.”





„Klassisches“ TDD



Die Regeln

- Schreibe nur Code, der verlangt wird.
- Entwickle schrittweise Deinen Code.
- Wähle möglichst kleine Schritte.
- Je allgemeingültiger der Code desto spezifischer der Test.



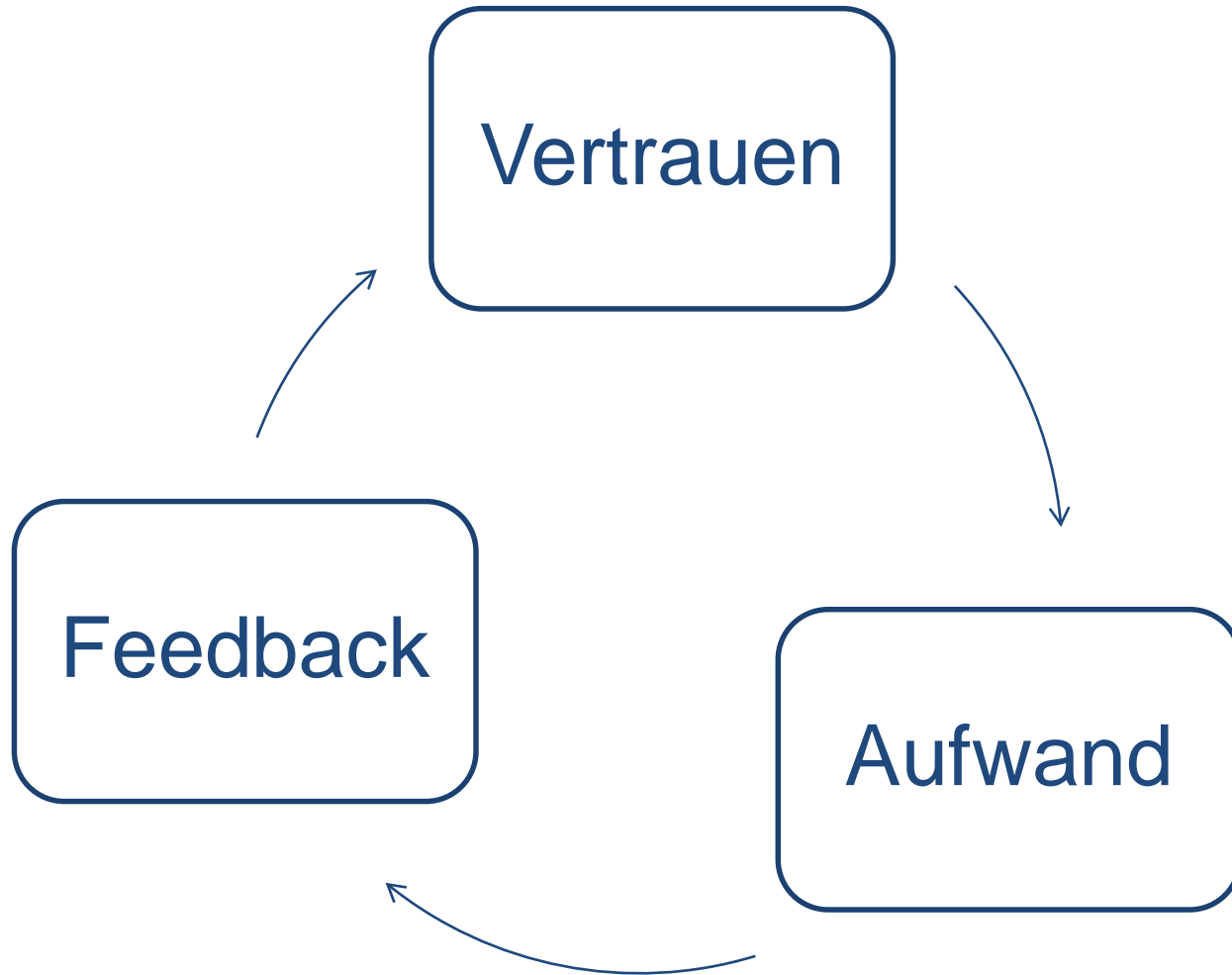
Transformation Priority Premise

({} → nil)	→ Keinen Code in Code umwandeln, der nichts implementiert.
(nil → constant)	→ Einführen einer Konstanten.
(constant → constant+)	→ Einfache Konstante in komplexere Konstante überführen.
(constant → scalar)	→ Konstante durch Variable oder Argument ersetzen.
(statement → statements)	→ Anweisung durch zusätzliche Anweisungen erweitern.
(unconditional → if)	→ Bedingungslose Codepfade in bedingte Codepfade ändern.
(scalar → array)	→ Zahlenwert in Array wandeln.
(array → container)	→ Array in eine Liste oder komplexeren Container überführen.
(statement → recursion)	→ Anweisung rekursiv umsetzen.
(if → while)	→ Fallunterscheidungen in Schleifen ändern.
(expression → function)	→ Ausdruck durch Funktion ersetzen.
(variable → assignment)	→ Variablen mit Zuweisungen erweitern.

DEMO!!!

„Pragmatisches“ TDD

DEMO!!!



1

Fast
Isolated
Independent
Repeatable
Self-Verifying
Timely

Pragmatisch

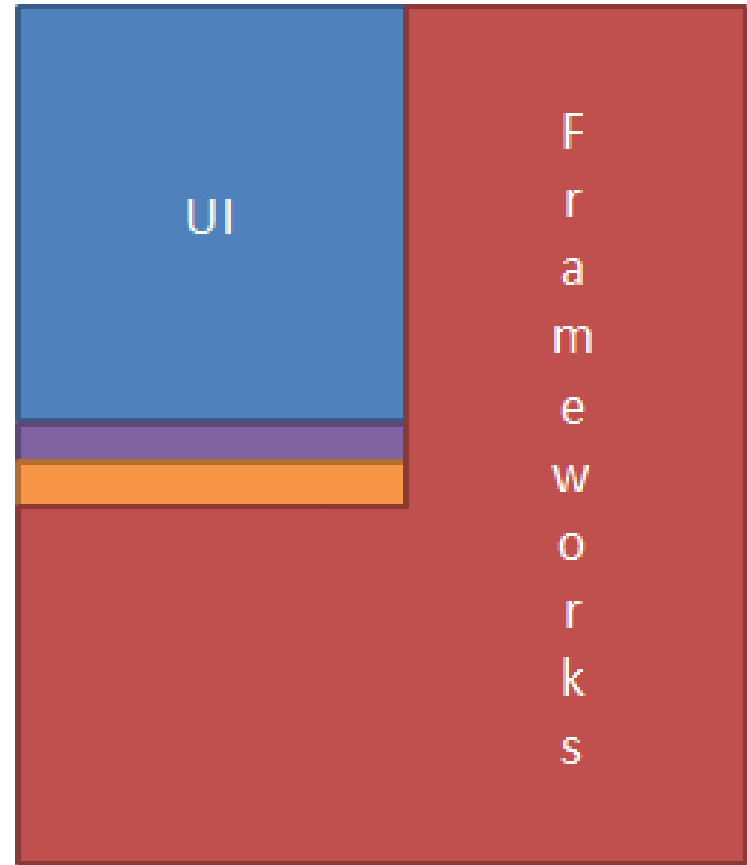
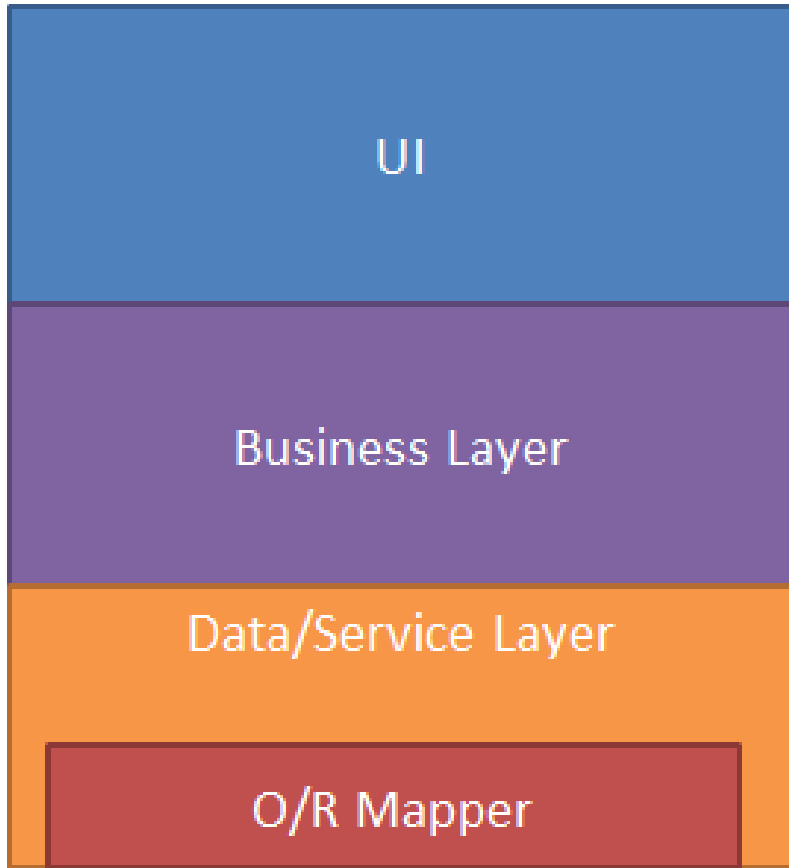
Eher Test First als Test Driven
Eher Komponententests als Unit-Tests
Eher interaktionsbasierend als
statusbasierend

```
public void Push(int value)
{
    Count++;
    content[Count] = value;
}
```

Klassisch

Eher Test Driven als Test First
Eher Unit-Tests als Komponententests
Eher statusbasierend als
interaktionsbasierend

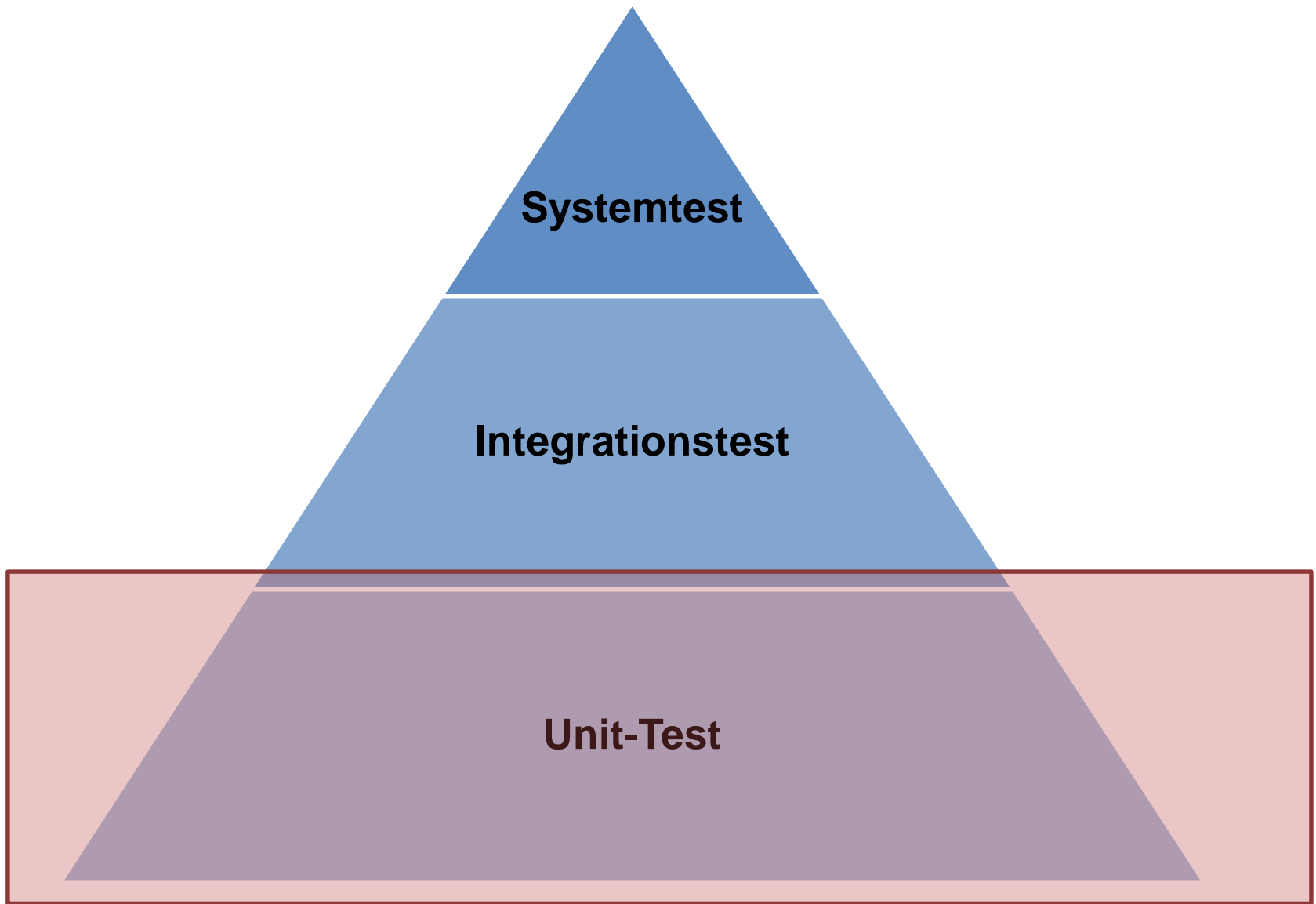
```
public void Push(int value)
{
    public void Push(int value)
        Count++; value;
        content[Count] = value;
}
```

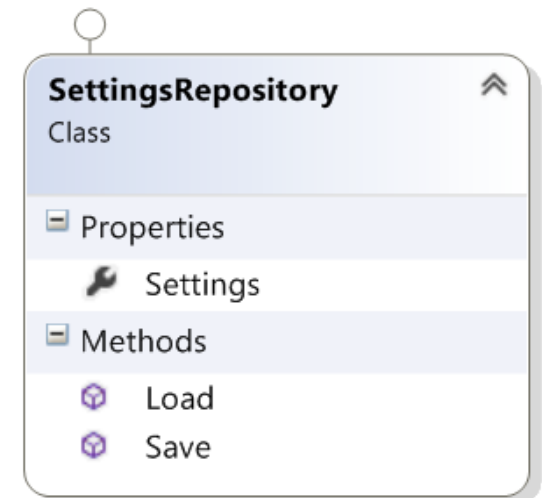
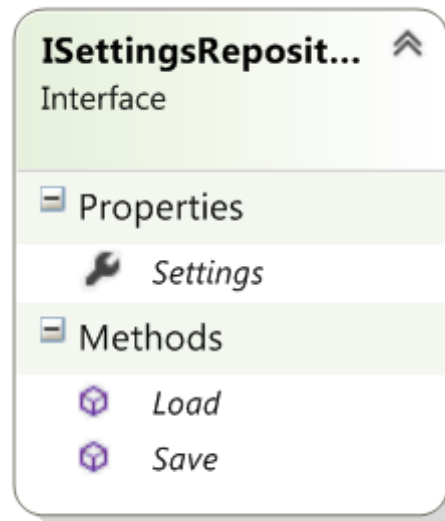
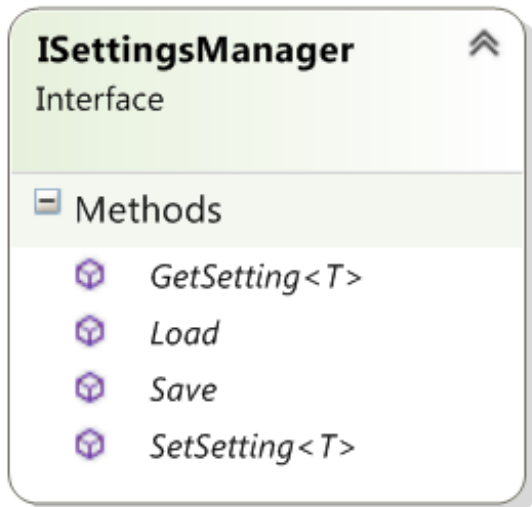



Red-Green-Implement



Fehlschlag

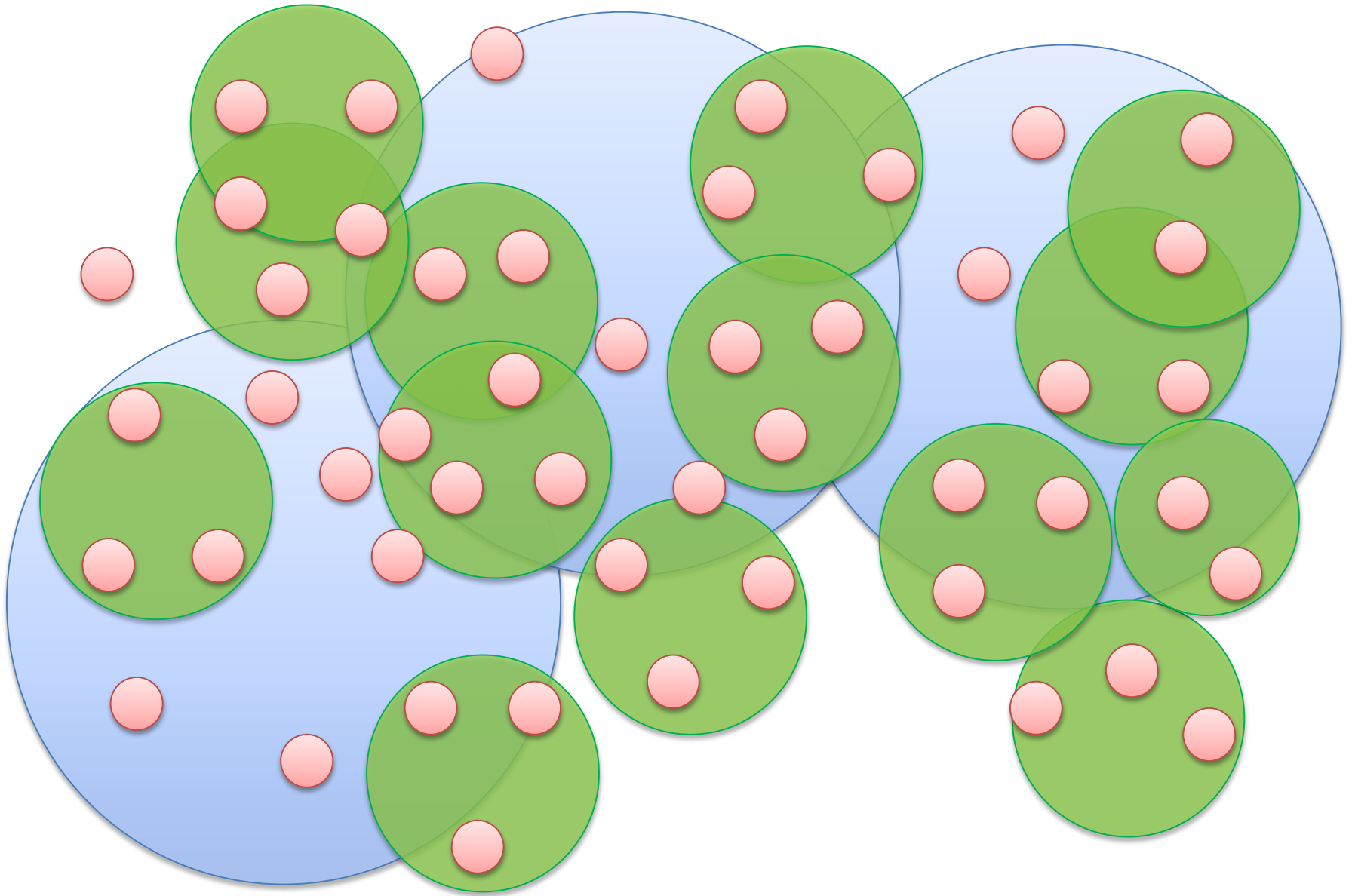


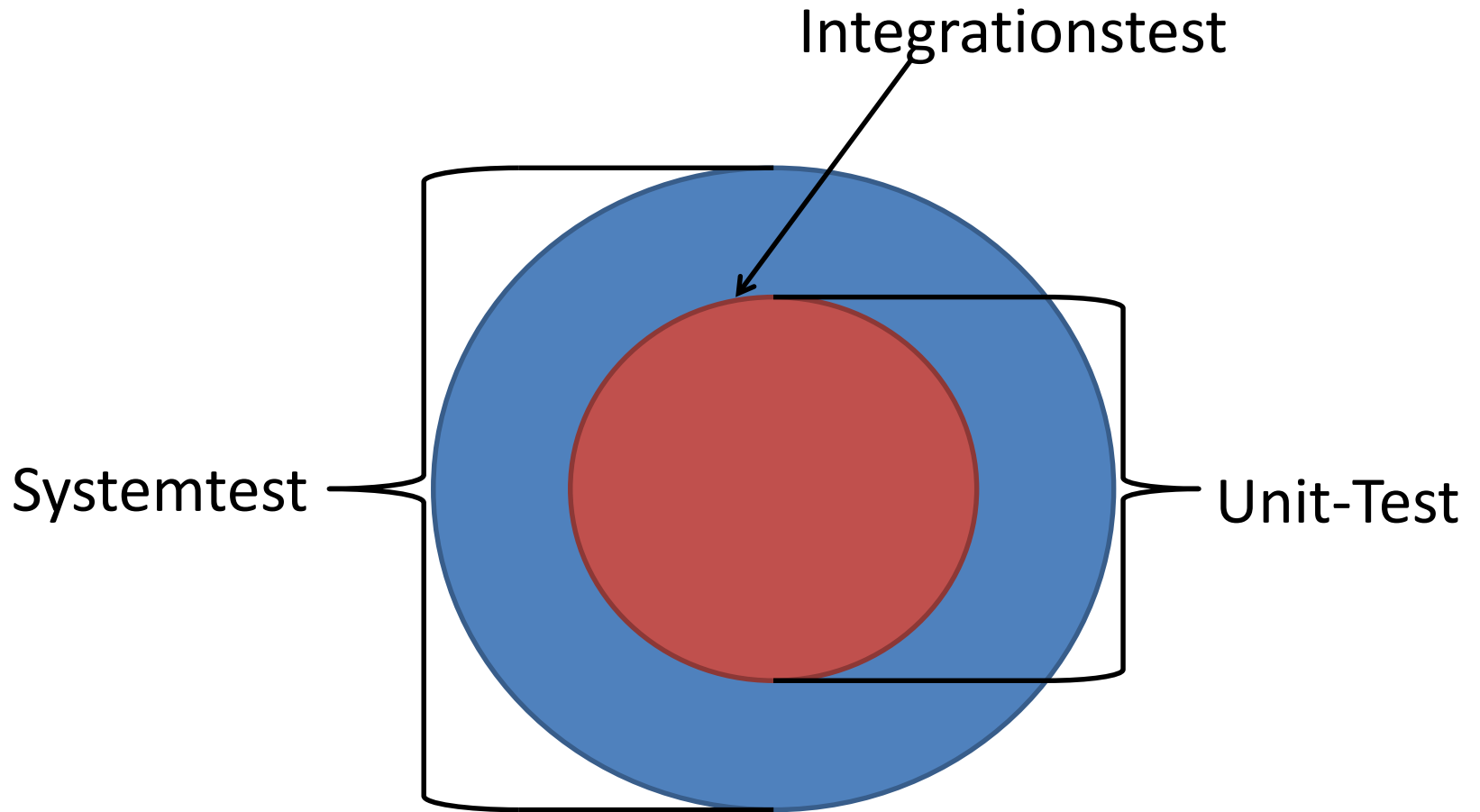


Unit-Test???

```
[TestMethod]
public void SettingOfType_String_ShallBeReadCorrectly()
{

}
}
```

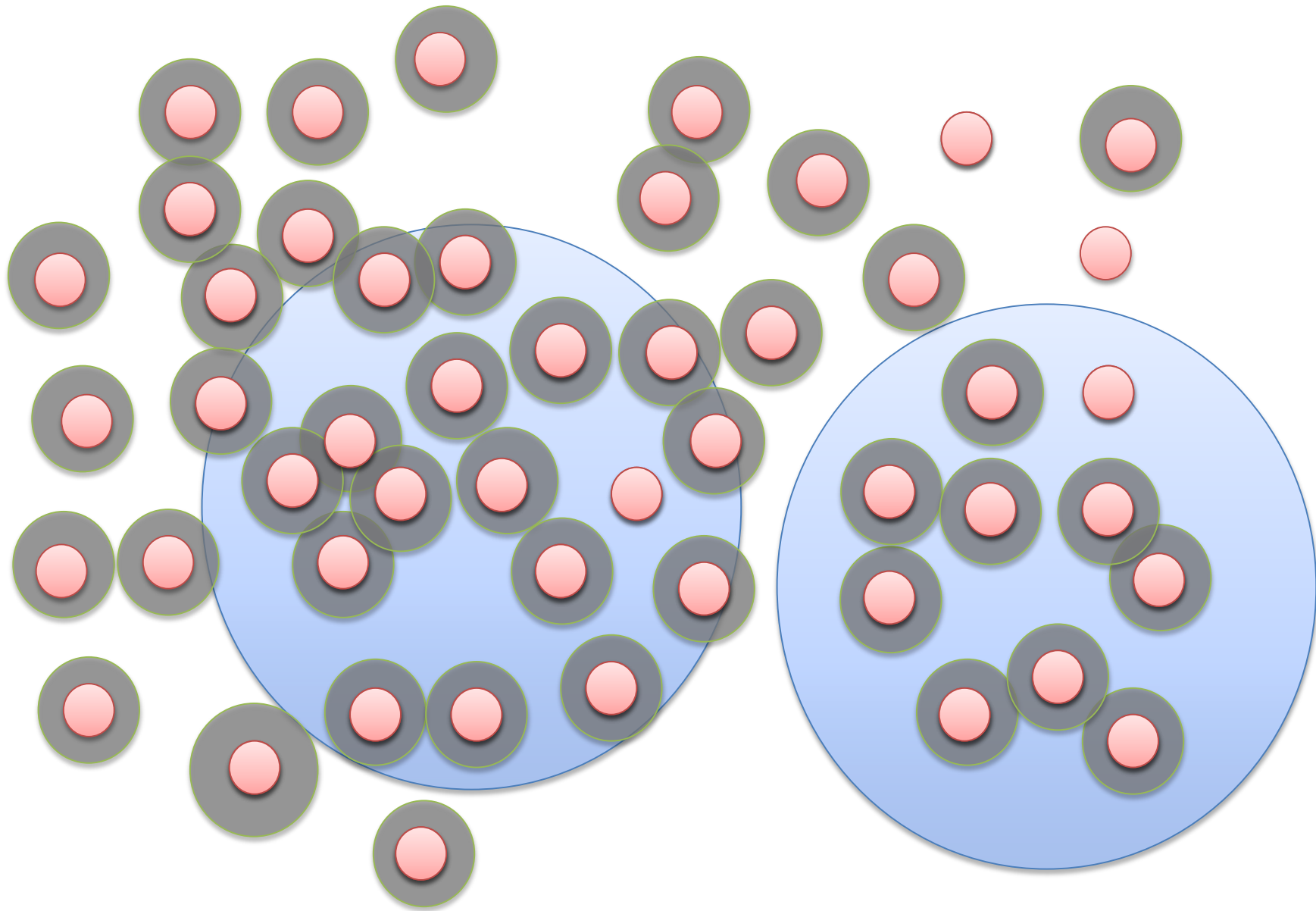


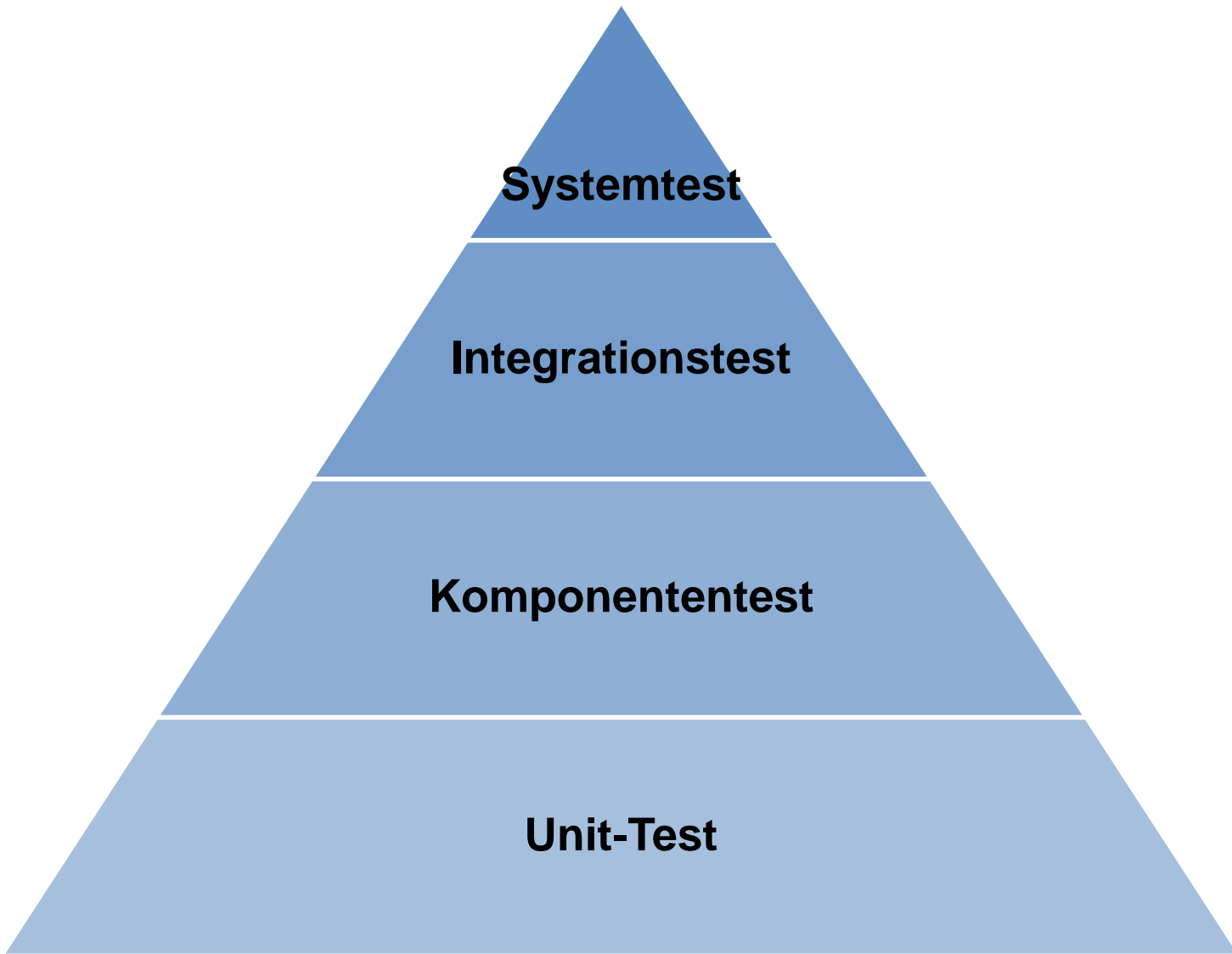


Unit-Test???

```
[TestMethod]  
public void SettingOfType_String_ShallBeReadCorrectly()  
{
```

```
}
```

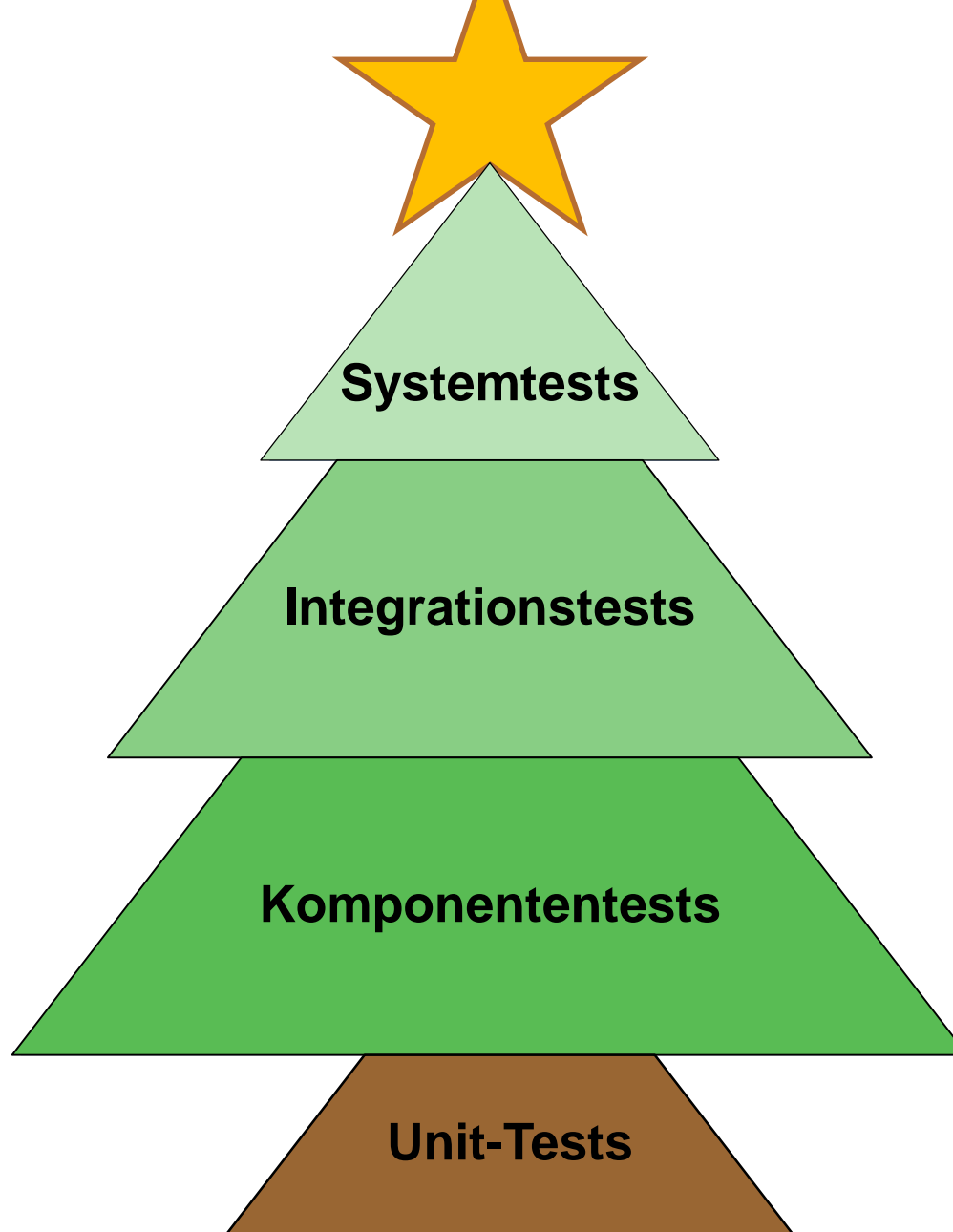


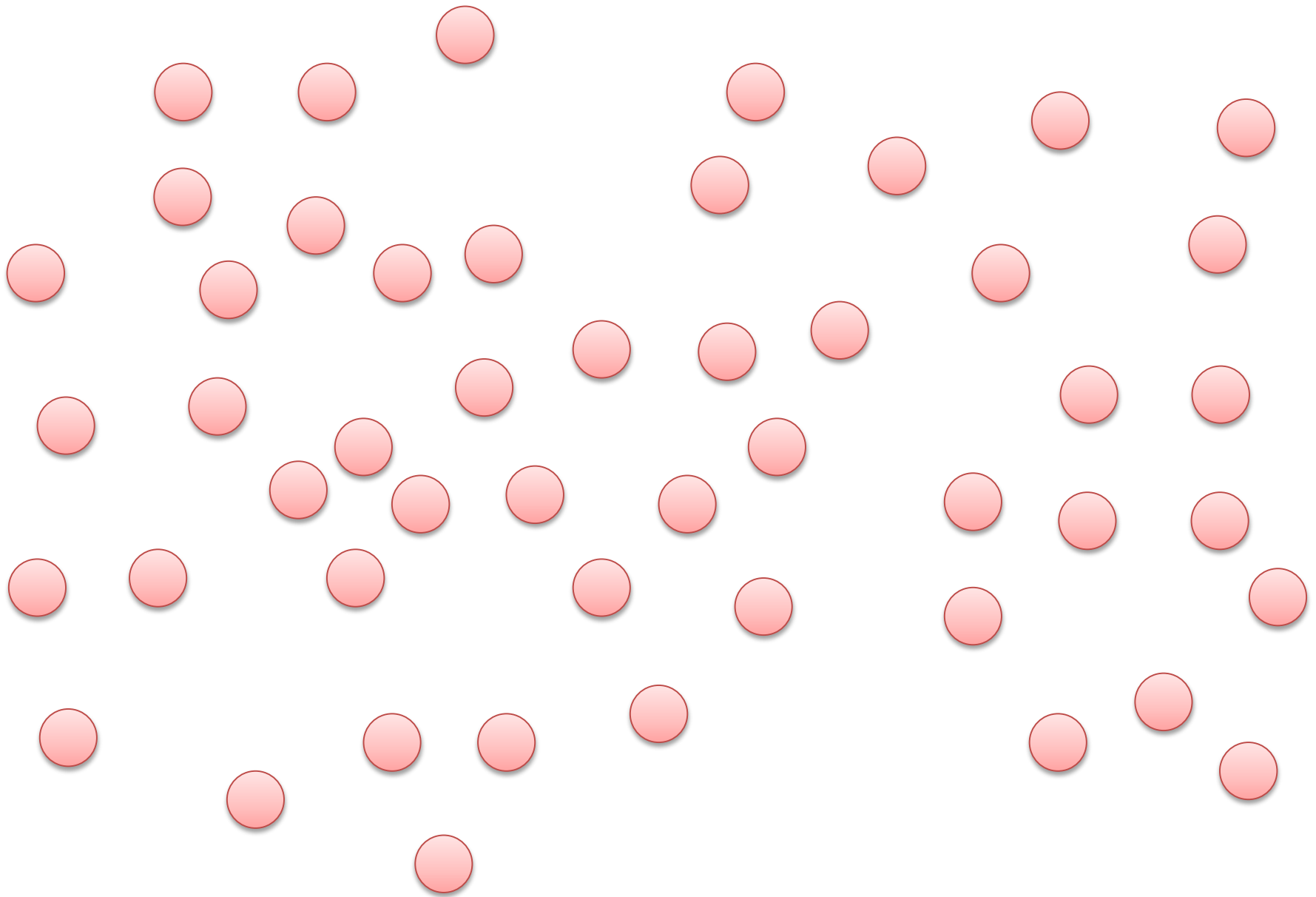
Systemtest

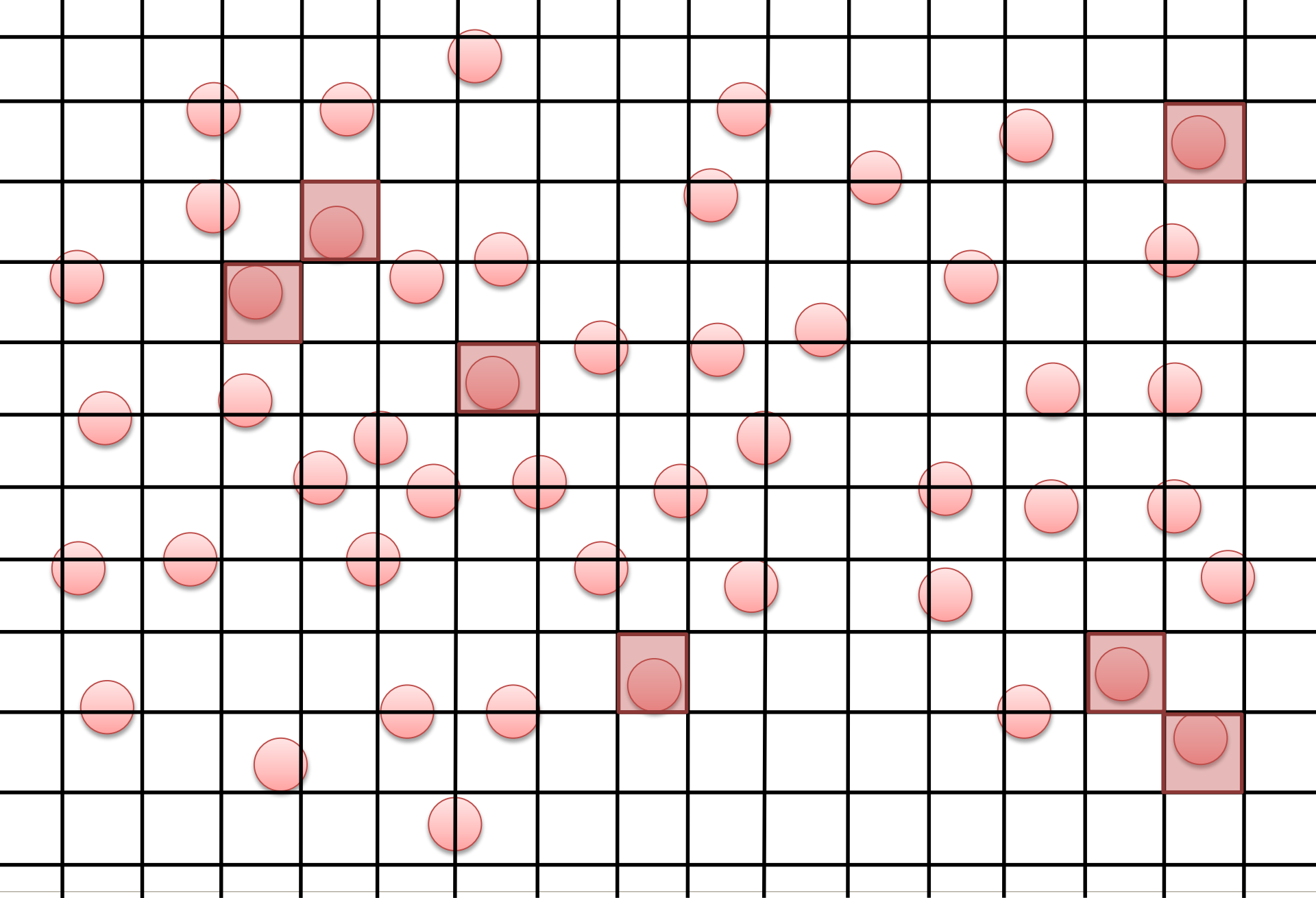
Integrationstest

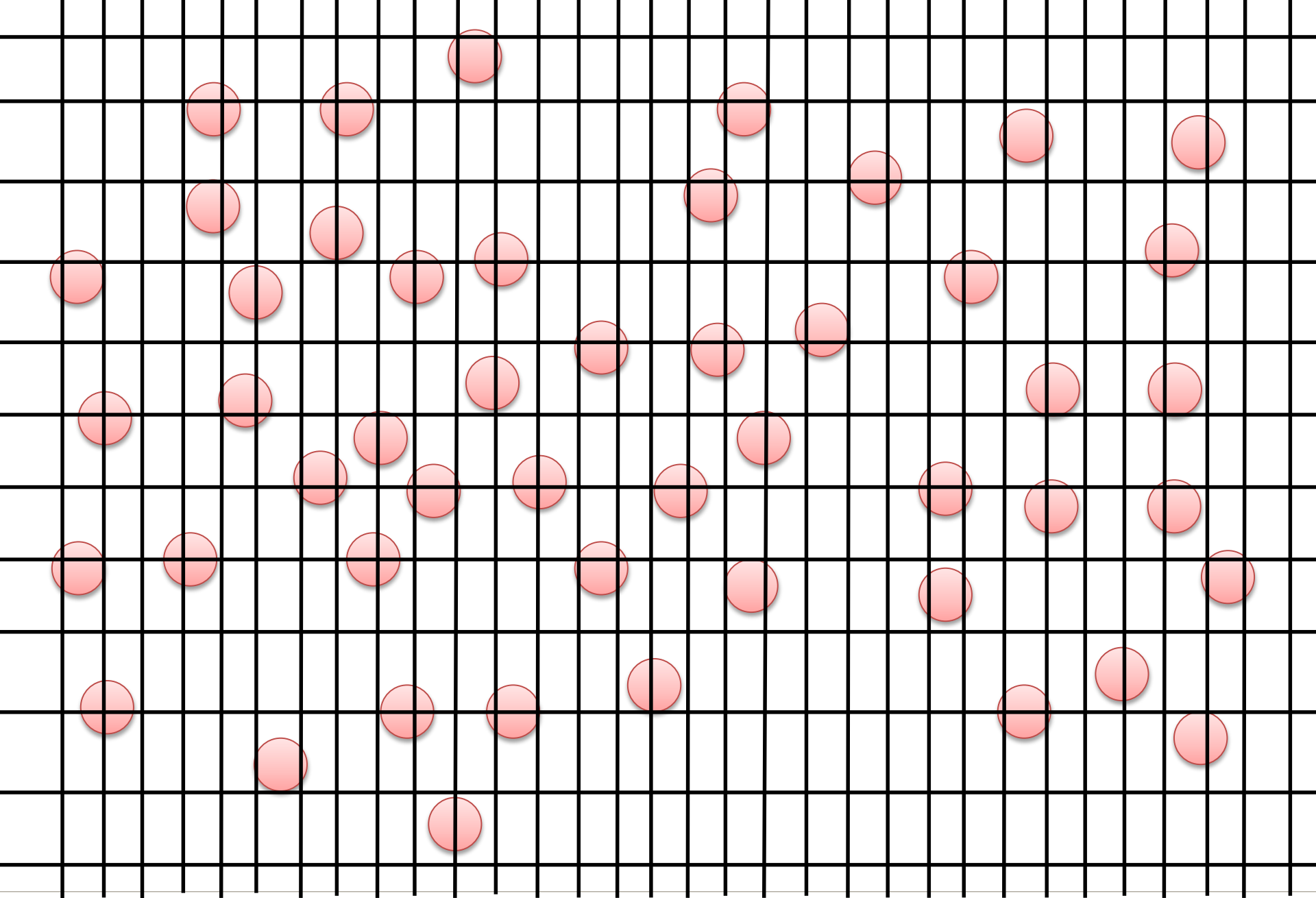
Komponententest

Unit-Test









Zusammenfassung