Software-Testing

Lektionen 9-12

Bildquelle: https://laptrinhx.com/topic/213/cac-dinh-nghia-va-thuat-ngu-trong-kiem-thu-phan-mem-phan-2

Zentrum für berufliche Weiterbildung

Agenda

- Repetition
- Vorgehensmodelle
- Fundamentaler Testprozess
- Psychologie des Testens
- Testcode Gestaltung
- AAA-Pattern
- Assertions

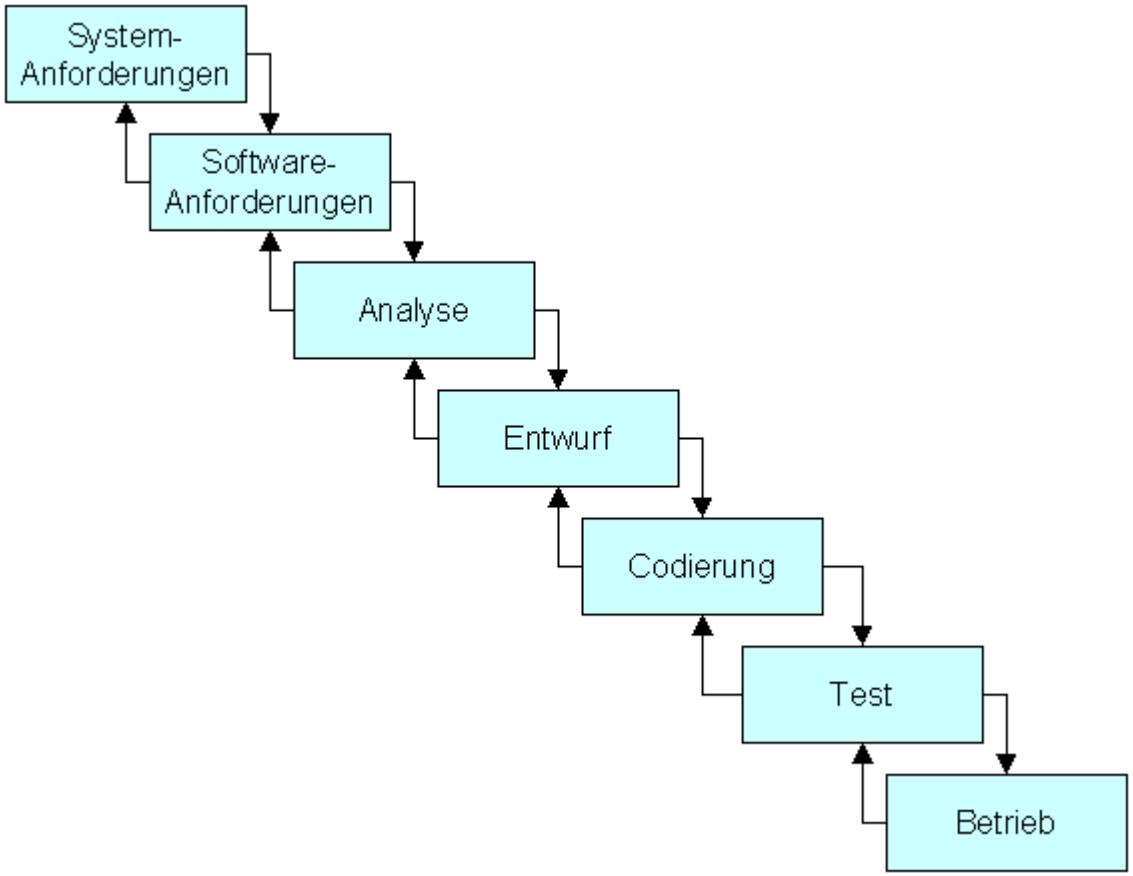
Repetition

- Drei Gruppen
- Drei Flipcharts
- Drei Themen
- Drei Mal vier Minuten
- Zwei Wechsel

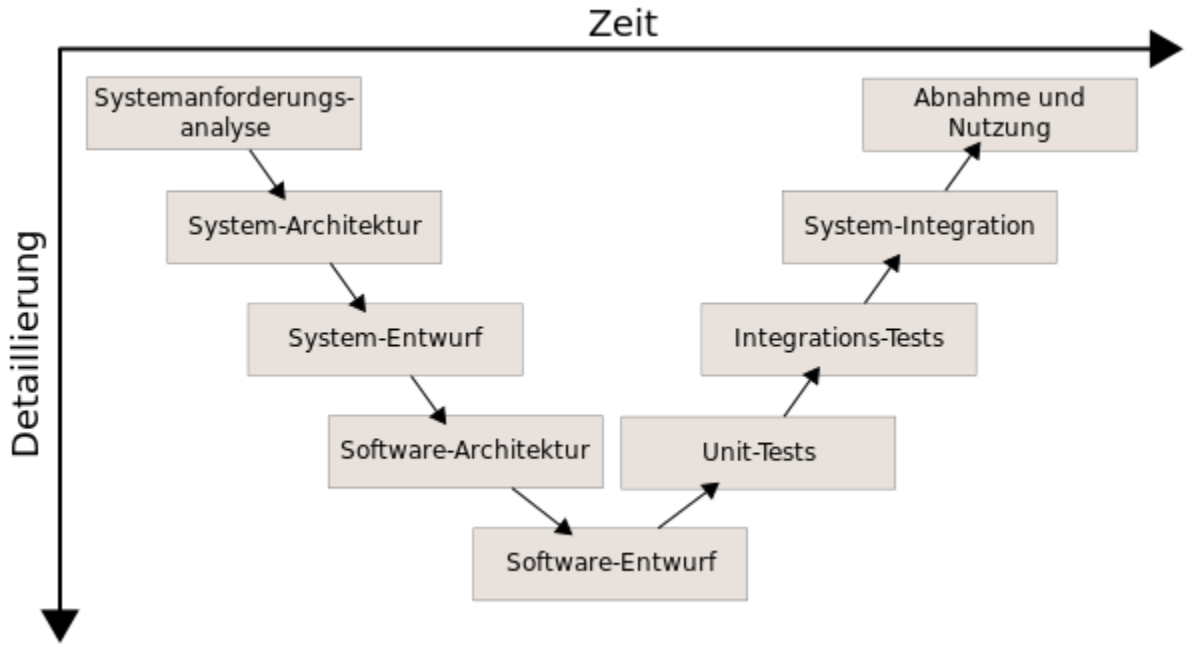
Vorgehensmodelle in der SW-Entwicklung

Bildquelle: http://www.sciencemag.org/features/2011/01/fix-system-not-women

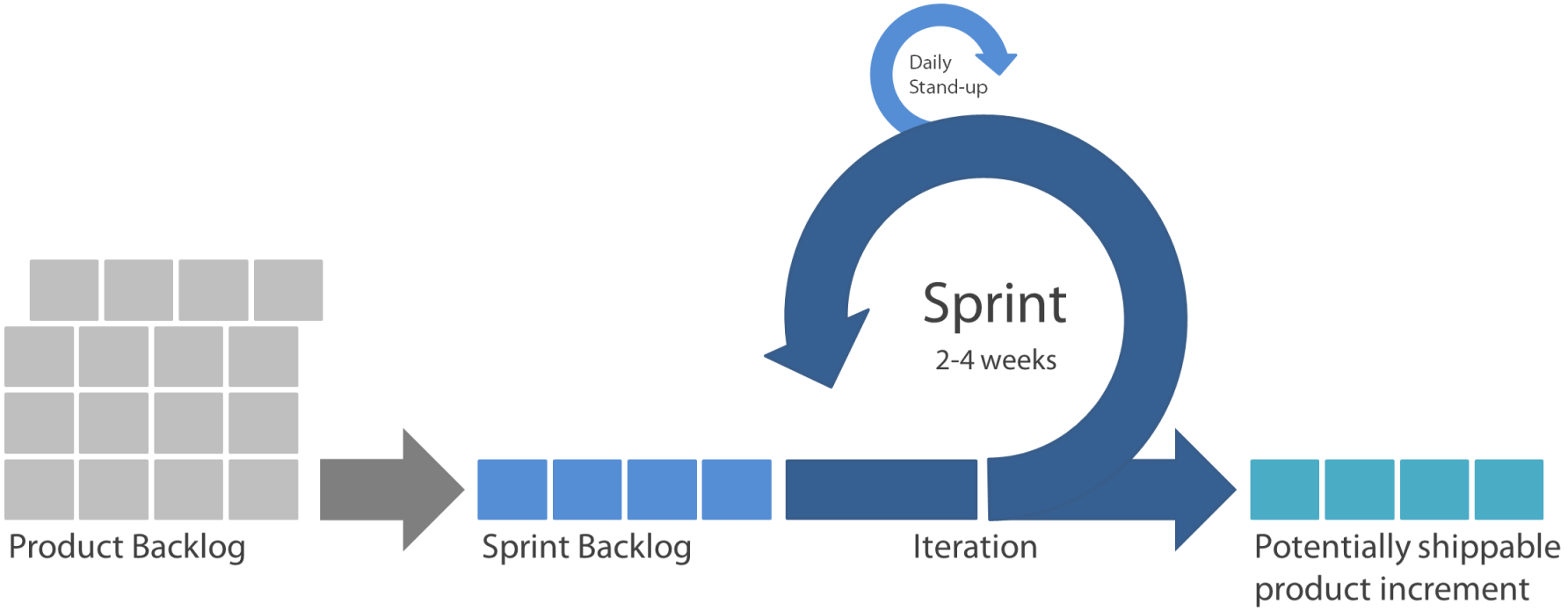
Vorgehensmodell (Wasserfall)



Vorgehensmodell (V-Modell)

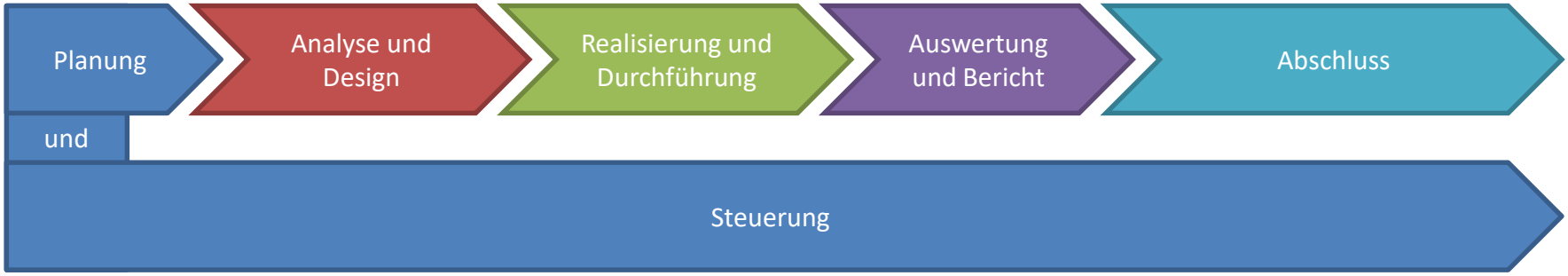


Vorgehensmodell (SCRUM)

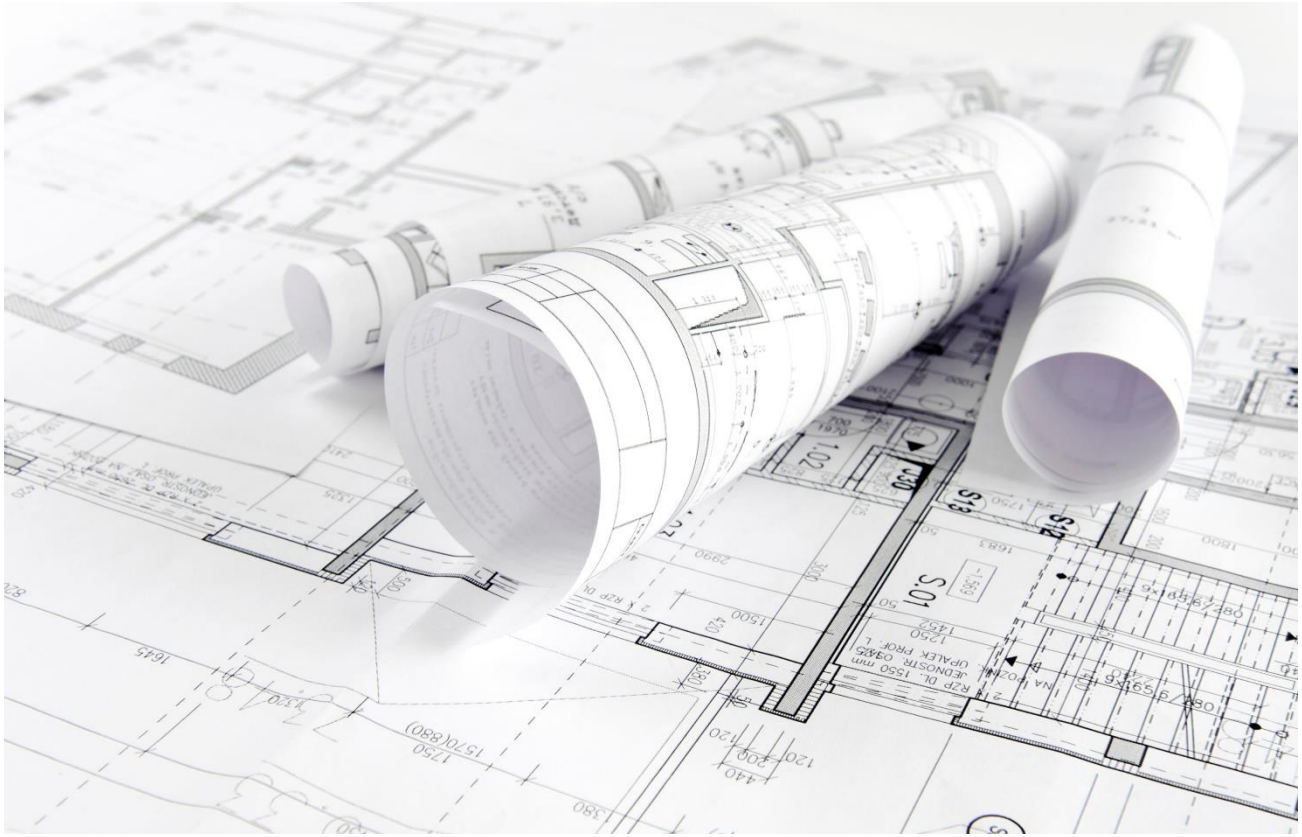


Zentrum für berufliche Weiterbildung

Fundamentaler Testprozess



Planung



Bildquelle: http://www.bplanung.ch/

Steuerung

Bildquelle: http://www.fls.de/de/produktionsplanung-und-steuerung-pps.htm

Testanalyse

Bildquelle: http://everyinvestor.co.uk/2015/05/06/how-to-analyse-a-structured-product/

Testdesign

Bildquelle: http://www.designandexecute.com/designs/good-vs-great-design/

Testrealisierung und Durchführung

Bildquelle: https://www.der-querschnitt.de/archive/14700

Testauswertung und Bericht

Bildquelle: http://www.rscaaretal.ch/?p=8887

Abschluss

Bildquelle: http://www.dw.com/bn/%E0%A6%AC%E0%A6%BE%E0%A6%B2%E0%A7%81%E0%A6%95%E0%A6%BE%E0%A6%AC%E0%A7%87%E0%A6%B2%E0%A6%BE%E0%A7%9F-%E0%A6%86%E0%A6%AE%E0%A6%BF-%E0%A6%B2%E0%A6%BF%E0%A6%96%E0%A7%87%E0%A6%9B%E0%A6%BF%E0%A6%A8%E0%A7%81/a-17261715

Übung «Fundamentaler Testprozess»

Bilden Sie drei Gruppen und diskutieren Sie zusammen über Analogien, resp. Unterschiede zwischen dem fundamentalen Testprozess und des Testprozesses in Ihrer Firma.

Fassen Sie Ihre Erkenntnisse zusammen und stellen Sie diese der Klasse vor.

Arbeitsform: Gruppenarbeit
Zeit: 20 Minuten
Besprechung / Feedback: In der Klasse

Psychologie des Testens

Bildquelle: http://12ghostwriters.com/ghostwriter/psychologie/

Prinzipien des Softwaretestens

- 1. Testen zeigt die Anwesenheit von Fehlern
- 2. Vollständiges Testen ist nicht möglich
- 3. Mit dem Testen frühzeitig beginnen
- 4. Häufung von Fehlern
- 5. Zunehmende Testresistenz
- 6. Testen ist abhängig vom Umfeld
- 7. Trugschluss: Keine Fehler bedeutet ein brauchbares System

Ethische Leitlinien nach ISTQB

- Öffentlichkeit
- Kunde und Arbeitgeber
- Produkt
- Urteilsvermögen
- Management
- Berufsbild
- Kollegen
- Persönliche Einstellung

Test-Code-Gestaltung nach Roy Osherove

- Benennung der Unit-Tests nach folgendem Schema:
  - [UnitOfWorkName]_[Szenario]_[ErwartetesVerhalten]
- z.B.
  - Addition_calculate_Returns5
  - Division_divideByZero_ThrowsException

Quelle: http://osherove.com/blog/2012/5/15/test-naming-conventions-with-unit-of-work.html

Name der Unit of Work (Was?)

- Welche Funktion eines SUT wird getestet?
- **Der Name der Unit of Work ist oft der Name der Methode die getestet werden soll, also ein Verb.**
- Beispiele:
  - Summate
  - Divide
  - Calculate
  - Init
  - usw.

Szenario (Wie?)

- Pro Unit of Work gibt es in der Regel mehrere Tests. Mit dem Szenario werden diese unterschieden.
- Beispiele:
  - Positiv
  - InvalidData
  - DivideByZero
  - usw.

Erwartetes Verhalten (Ergebnis)

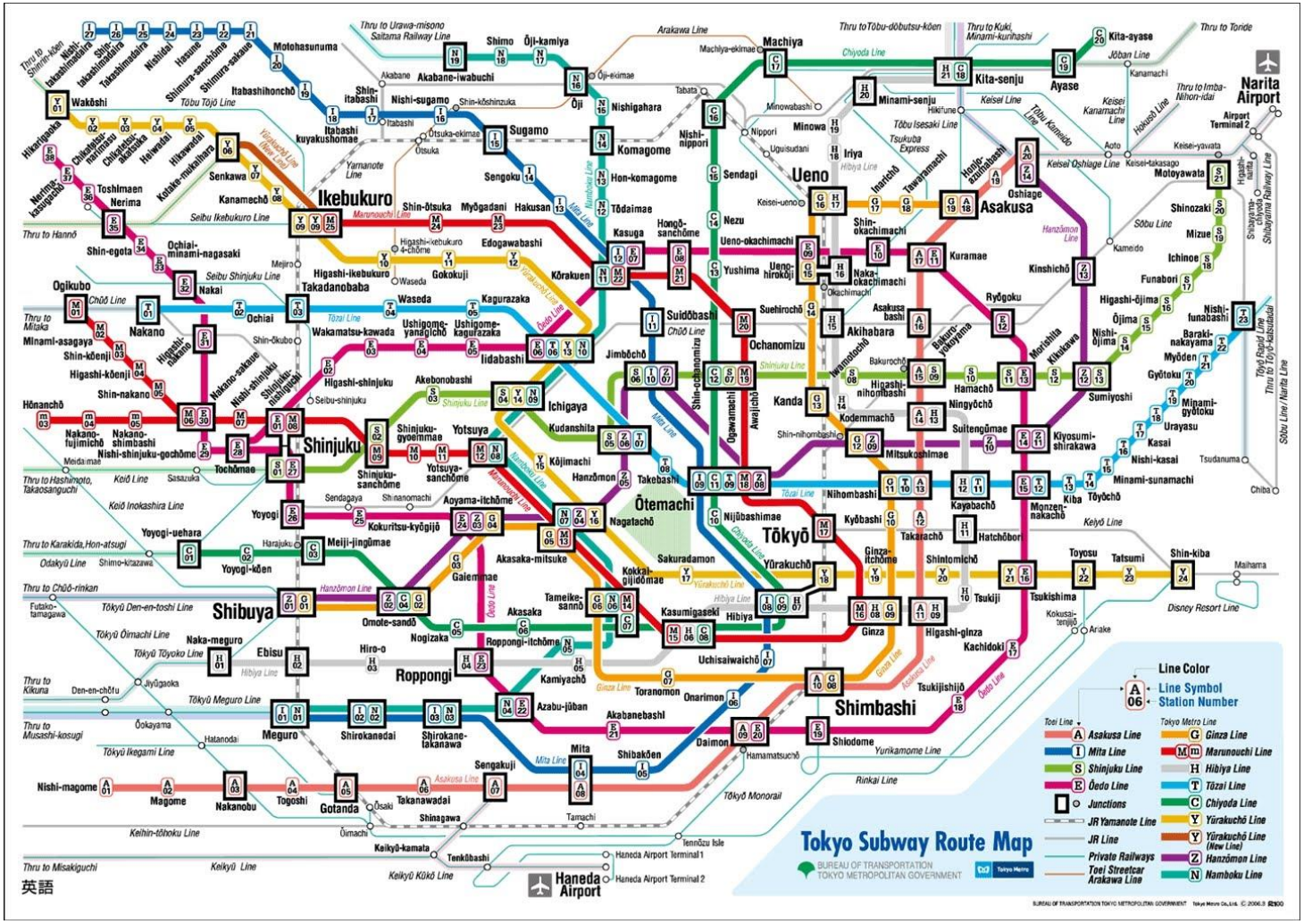
- Definiert, wie das Resulatat des Tests aussehen soll
- Beispiele:
  - ReturnsCorrectResult
  - ThrowsException
  - UserExists
  - WriteToConsole
  - usw.

Drei mögliche Ergebnisse

- **Rückgabe eines Wertes** (echter Wert oder Exception)
  - `IsLoginOK_UserDoesNotExist_ReturnsFalse`
- **Änderung des Systemzustandes**
  - `AddUser_ValidUserDetails_UserCanBeLoggedIn`
- **Aufruf einer anderen Komponenten**
  - `IsLoginOK_LoginFails_CallsLogger`

Quelle: http://osherove.com/blog/2012/5/15/test-naming-conventions-with-unit-of-work.html

Übersichtlich?



Bildquelle: http://vivianintokyo.blogspot.com/2012/10/tokyo-von-oben-und-unten.html

Zentrum für berufliche Weiterbildung

Patterns für Methodenrumpf

- Given-When-Then (GWT)
  - Gut für Behavior Driven Development
  - Von Dan North und Chris Matts
- Setup-Exercice-Verify-Teardown
  - Gut für Unit Tests und Integration Tests
- AAA-Pattern
  - Gut für Unit Tests
  - Bill Wake

Setup-Exercice-Verify-Teardown

- Setup
  - Alles nötige Bereitstellen inkl. dem SUT
- Exercise
  - Aktion ausführen und das Resultat in eine Variabel speichern
- Verify
  - Überprüfen, ob das Ergebnis korrekt ist
- Teardown
  - Ressourcen des Tests abbauen

AAA-Pattern

- **Arrange (anordnen)**
  - Alles nötige Bereitstellen inkl. dem SUT
- **Act (die Tat)**
  - Aktion ausführen und das Resultat in eine Variabel speichern
- **Assert (behaupten)**
  - «Das müsste passieren…»

Ein Beispiel

```
[Test]
public void Division_Calculate_ReturnsQuotient()
{
    // arrange
    var basicOperation = new BasicOperation();

    // act
    var result = basicOperation.Division(100, 5);

    // assert
    Assert.That(result, Is.EqualTo(20));
}
```

Assertions | Behauptungen


```
Assert.AreEqual(4, 2+2);
Assert.That(2+2, Is.EqualTo(4));
```

Classic Model

- Wird nicht mehr weiterentwickelt…
  - Assert.True
  - Assert.False
  - Assert.Null
  - Assert.NotNull
  - Assert.Zero
  - Assert.NotZero
  - Assert.IsNaN
  - Assert.IsEmpty
  - Assert.IsNotEmpty
  - Assert.AreEqual
  - Assert.AreNotEqual
  - Assert.AreSame
  - Assert.AreNotSame
  - Assert.Contains
  - Assert.Greater
  - Assert.GreaterOrEqual
  - Assert.Less
  - Assert.LessOrEqual
  - Assert.Positive
  - Assert.Negative
  - Assert.IsInstanceOf
  - Assert.IsNotInstanceOf
  - Assert.IsAssignableFrom
  - Assert.IsNotAssignableFrom
  - Assert.Throws
  - Assert.ThrowsAsync
  - Assert.DoesNotThrow
  - Assert.DoesNotThrowAsync
  - Assert.Catch
  - Assert.CatchAsync
  - Assert.Pass
  - Assert.Fail
  - Assert.Ignore
  - Assert.Inconclusive

Constraint Model

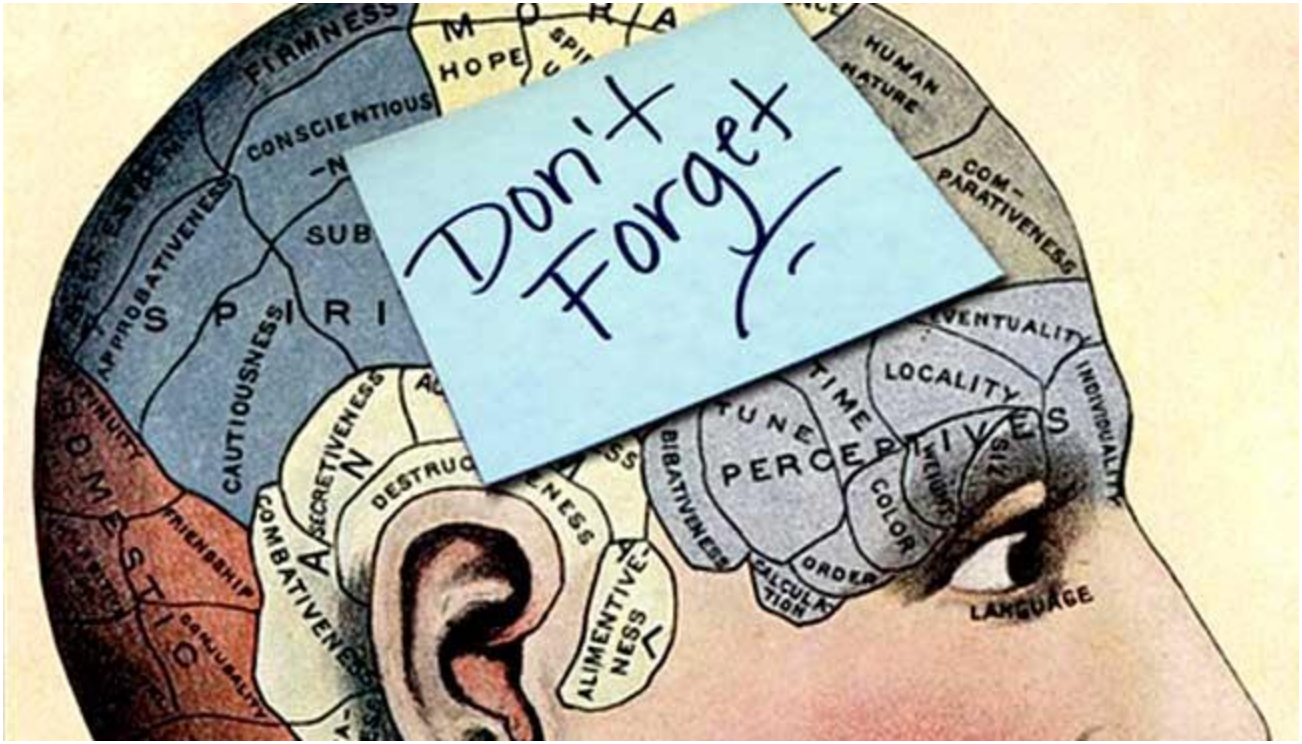
```
int[] array = new int[] { 1, 2, 3 };
Assert.That(array, Has.Exactly(1).EqualTo(3));
Assert.That(array, Has.Exactly(2).GreaterThan(1));
Assert.That(array, Has.Exactly(3).LessThan(100));
```

Übung «MathLibrary.Extended»

- Erstellen Sie Unit Tests für die MathLibrary.Extended.
- Source Code unter:
  - https://github.com/michikeiser/ZbW.Testing.MathExtended

Arbeitsform: Einzelarbeit
Zeit (Vorbereitung): Bis fertig
Besprechung / Feedback: Lösungen vorh.

Was nehmen wir vom Unterricht mit?



Bildquelle: http://floraremedia.com/keep-mind-memory-sharp/

Hausaufgaben

- Übung fertig machen
- nächstes Mal ca. 200 Zeilen Code für eine Code Review mitbringen

Ergänzende Unterlagen

Code Beispiele des heutigen Unterrichts unter:

https://github.com/michikeiser/ZbW.Testing.MathExtended