MATHEMA

Saubere Sache – aber wie?

Michael Wiedeking

Clean Code Days, 19. – 21. 6. 2018

Was überhaupt ist „Clean Code“?

- Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber“ bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann. Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.

- Wikipedia

- Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber“ bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann. Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.

- Wikipedia

- Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber" bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. **Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann.** Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.

- Wikipedia

- Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber" bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann. Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.

- Wikipedia

- Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber" bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann. Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.
- Wikipedia

Was eigentlich ist „Clean Code“?

- I like my code to be elegant and efficient. The logic should be straightforward to make it hard for bugs to hide, the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so not to tempt people to make messy with unprincipled optimizations. Clean code does one thing well.
- Bjarne Stroustrup

Was eigentlich ist „Clean Code“?

- Clean Code is simple and direct. Clean code reads like well-written prose. Clean code never obscures the designer’s intent but rather is full of crisp abstractions and straightforward lines of control.
- Grady Booch

Was eigentlich ist „Clean Code“?

- Clean code can be read, and enhanced by a developer other than its original author. It has unit and acceptance tests. It has meaningful names. It provides one way rather than many ways for doing one thing. It has minimal dependencies, which are explicitly defined, and provides a clear and minimal API. Code should be literate since depending on the language, not all necessary information can be expressed clearly in code alone.
- Dave Thomas

- I could list all the qualities that I notice in clean code, but there is one overarching quality that leads to all of them. Clean code always looks like it was written by someone who cares. There is nothing obvious that you can do to make it better. All of those things were thought about by the code's author, and if you try to imagine improvements, you're led back to where you are, sitting in appreciation of the code someone left for you—code left by someone who cares deeply about the craft.
- Michael Feathers

- In recent years I begin, and nearly end, with Beck’s rules of simple code. In priority order, simple code:
  - Runs all the tests;
  - Contains no duplication;
  - Expresses all the design ideas that are in the system;
  - Minimizes the number of entities such as classes, methods, functions, and the like.

  [...]
- Ron Jeffries

Was eigentlich ist „Clean Code“?

- You know you are working on clean code when each routine you read turns out to be pretty much what you expected. You can call it beautiful code when the code also makes it look like the language was made for the problem.
- Ward Cunningham

- Good Programming is not learned from generalities, but by seeing how significant programs can be made clean, easy to read, easy to maintain and modifiy, human engineered, efficient, and reliable, by the application common sense and good programming practices. Careful study and imitation of good programs leads to better writing.
- Brian W. Kernighan and P. J. Plauger
- Software Tools (1976)

Was überhaupt ist „Clean Code“?

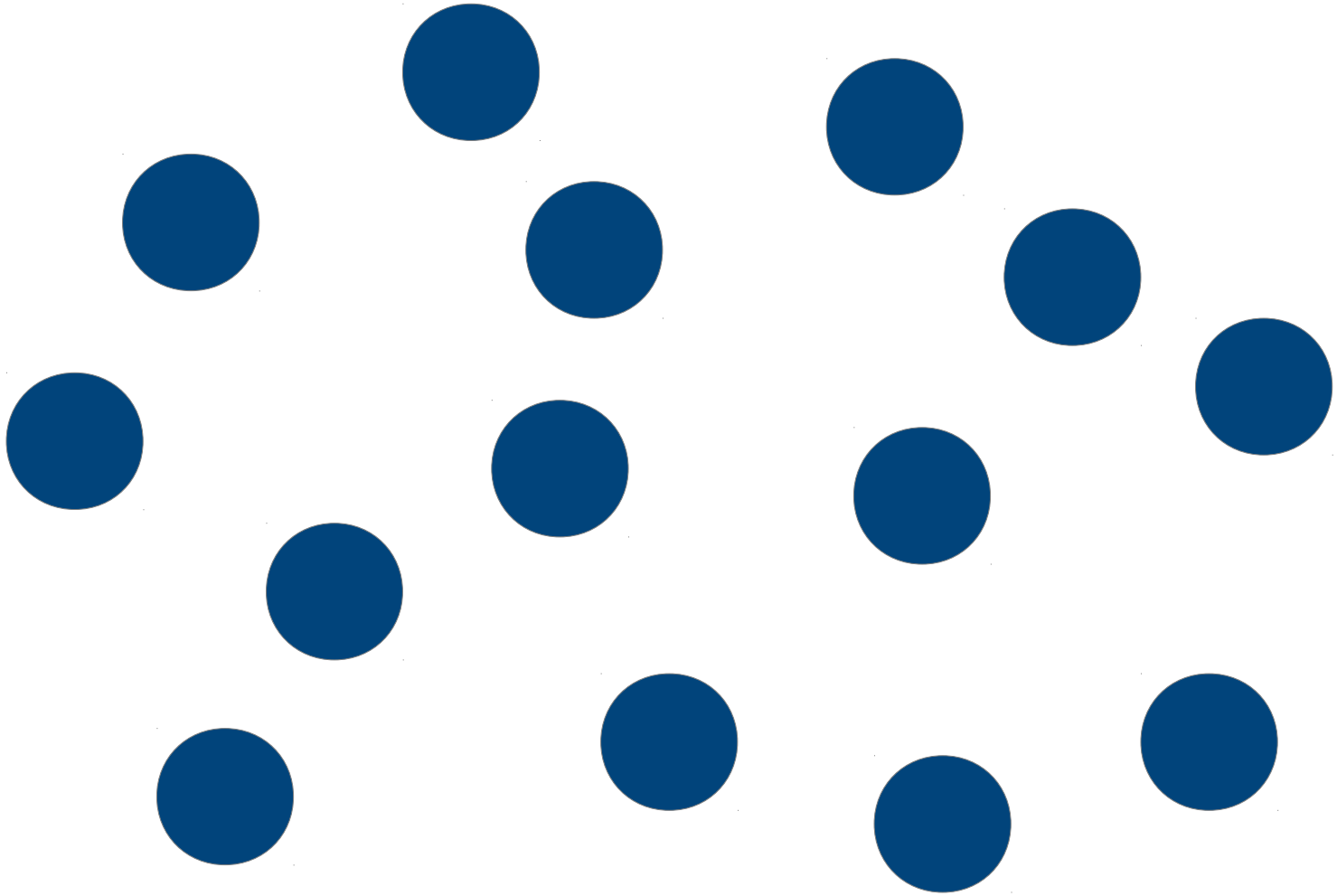
Was immer „Uncle Bob“ darunter versteht!

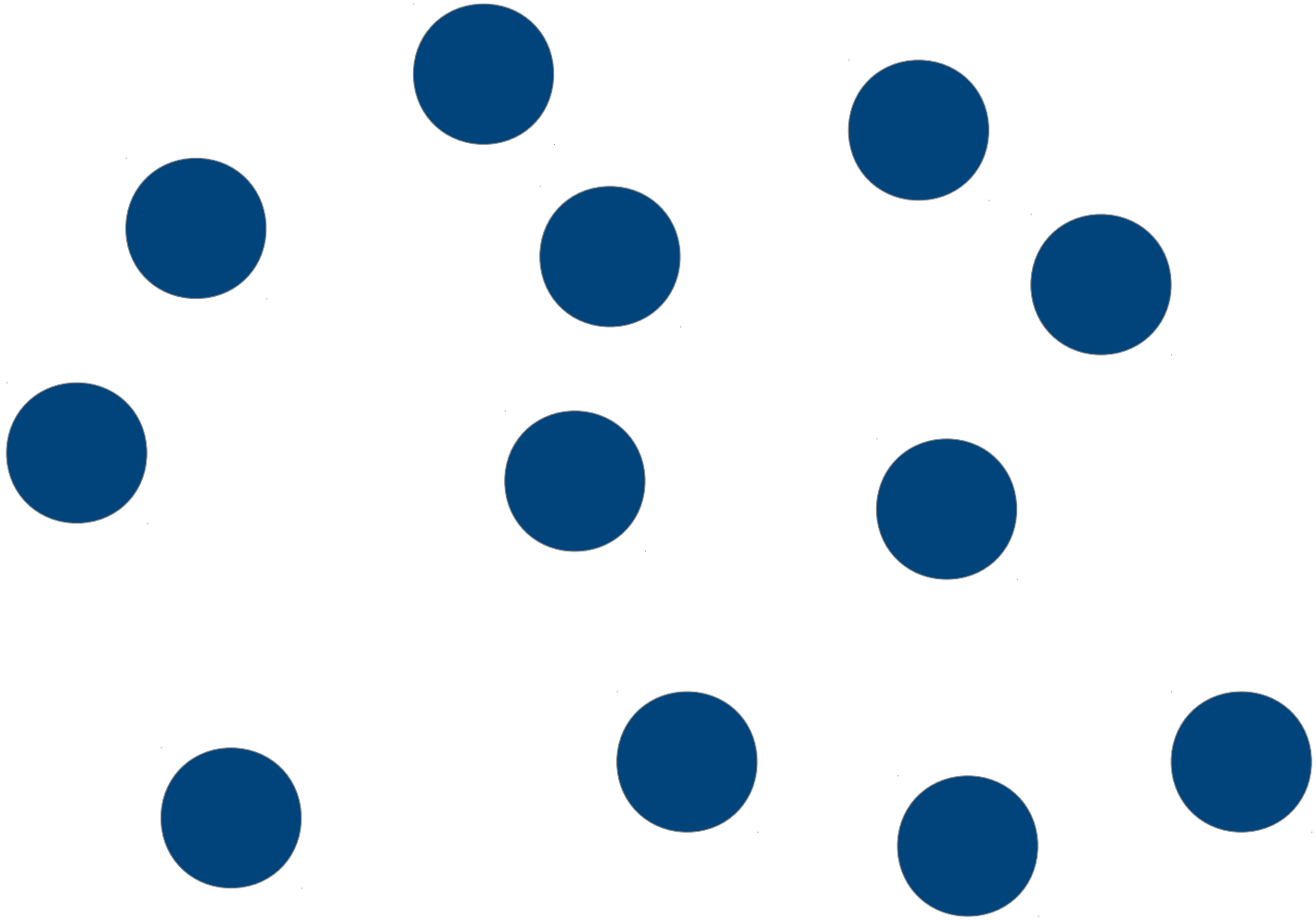
Für jeden etwas anderes!

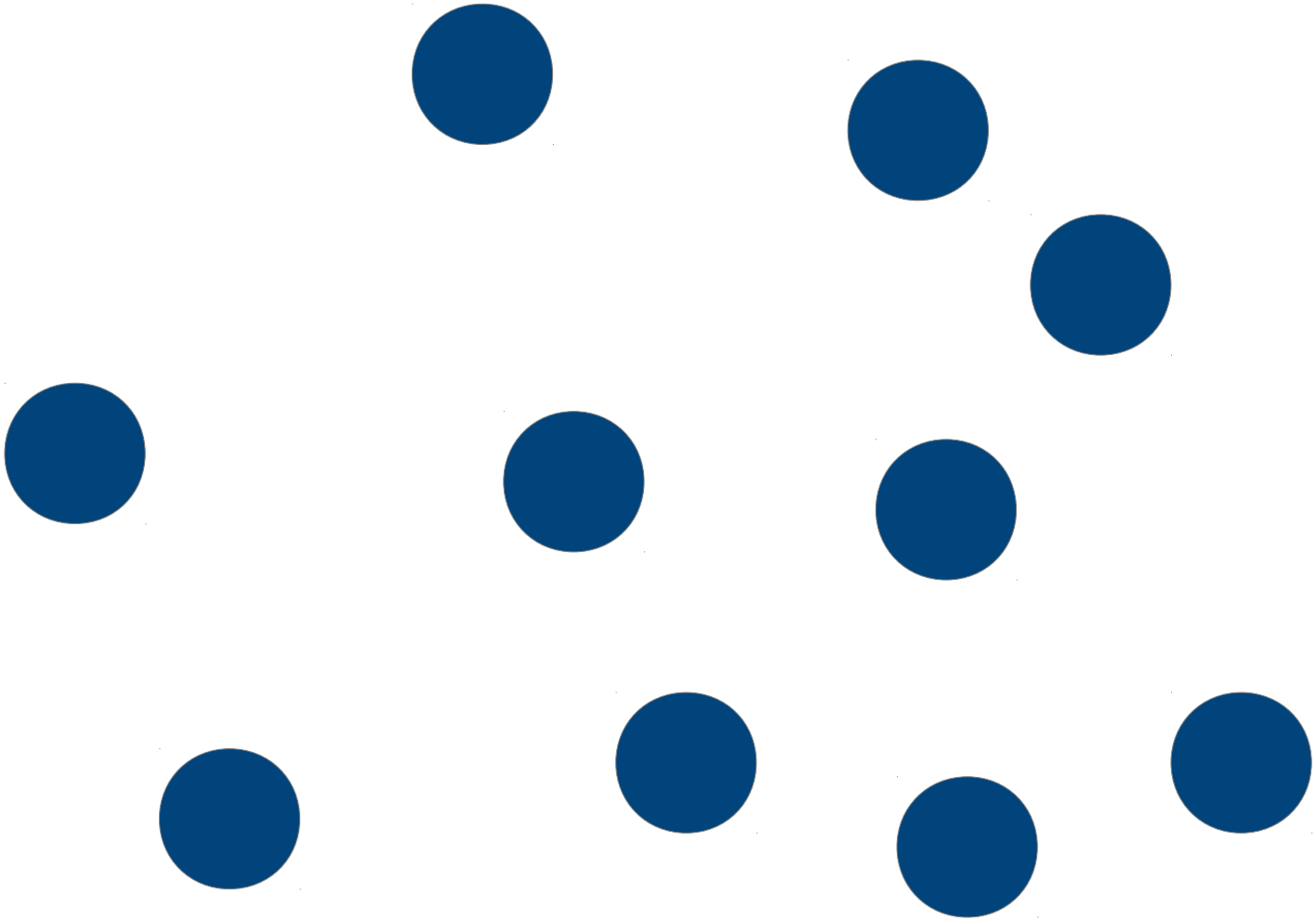
Das, was die anderen darunter verstehen!

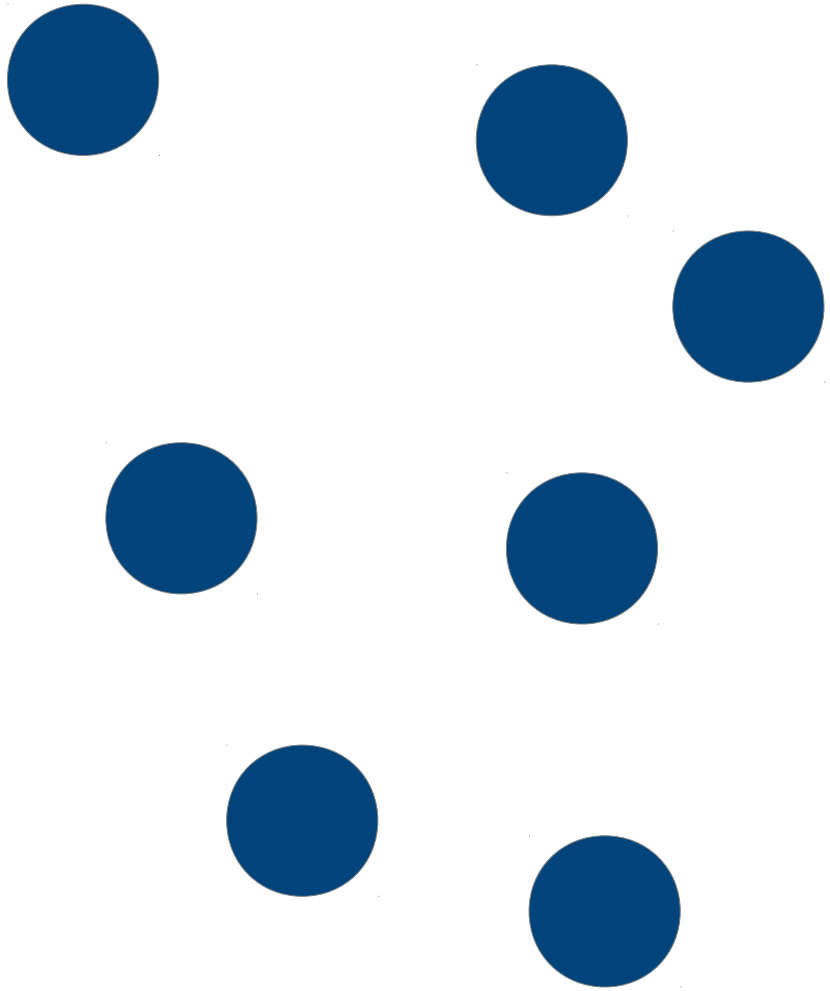
Das, was die anderen* darunter verstehen!

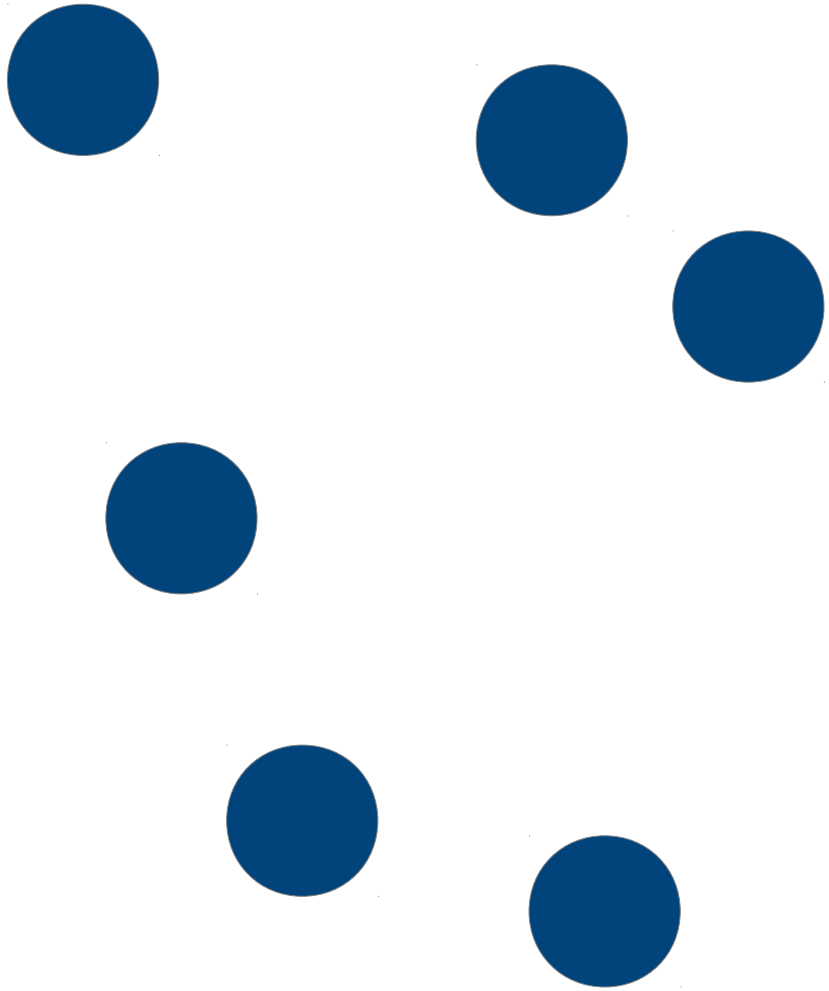
Was bedeutet dies nun konkret?

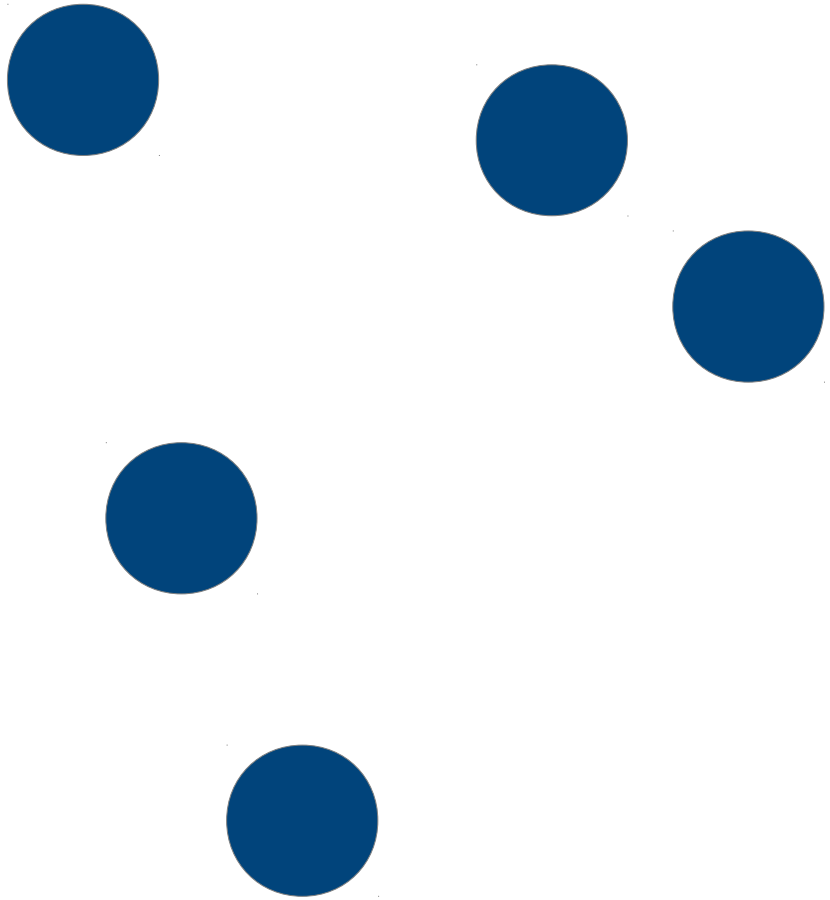


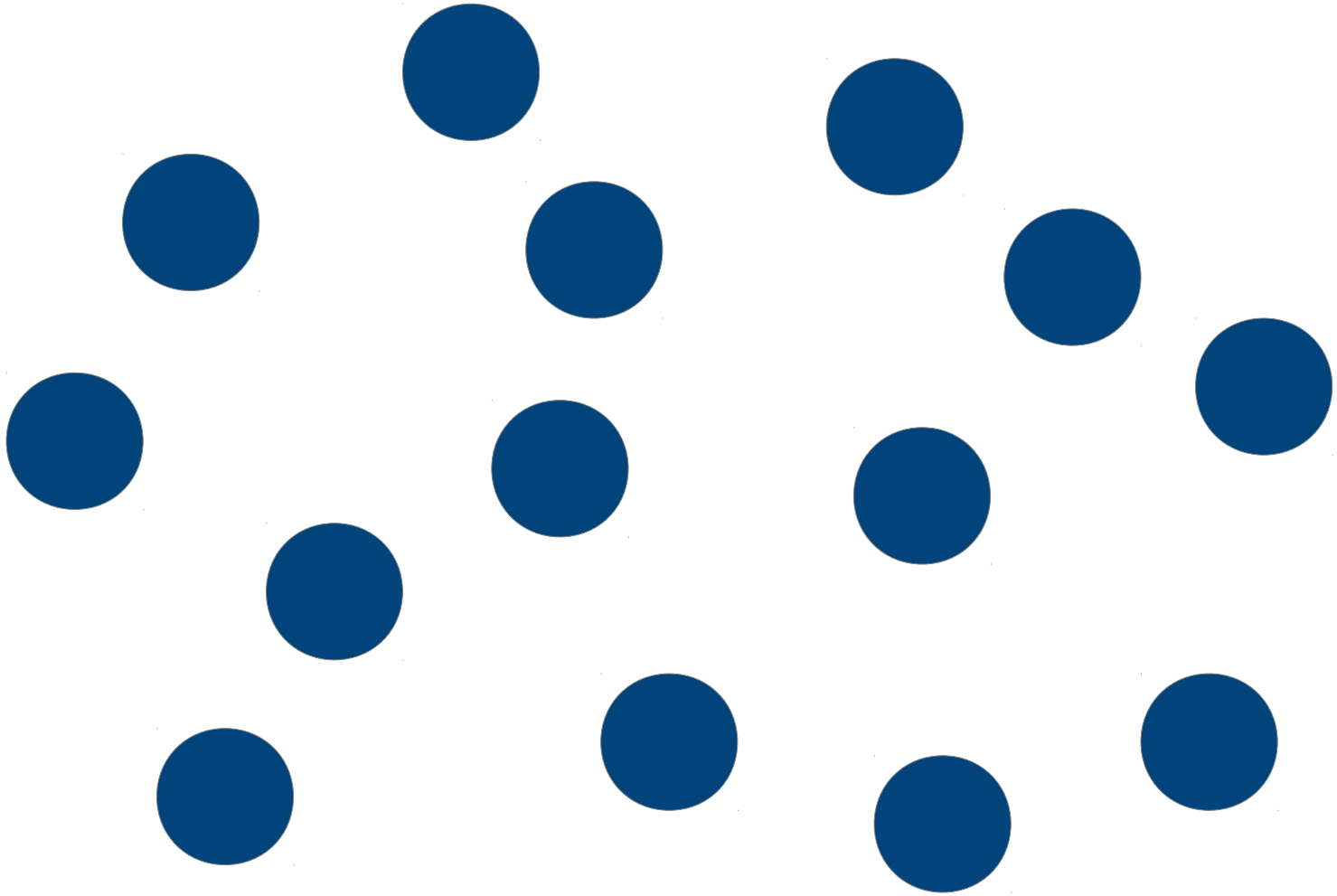


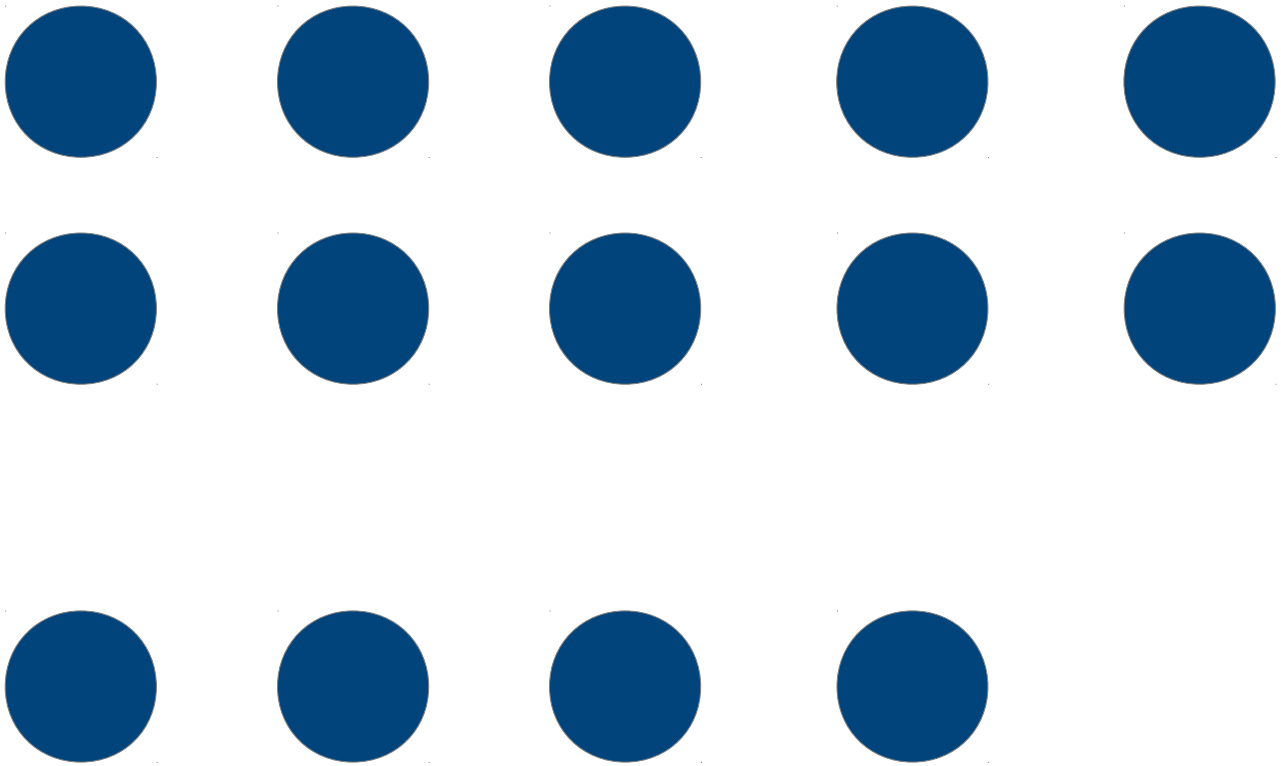


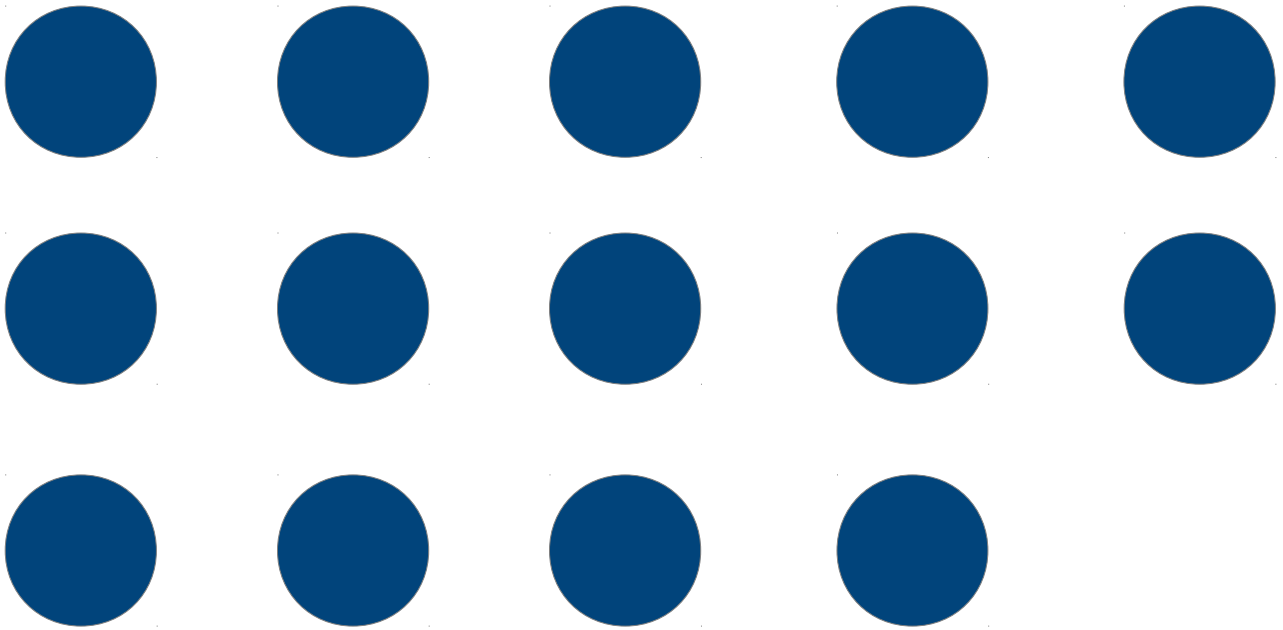


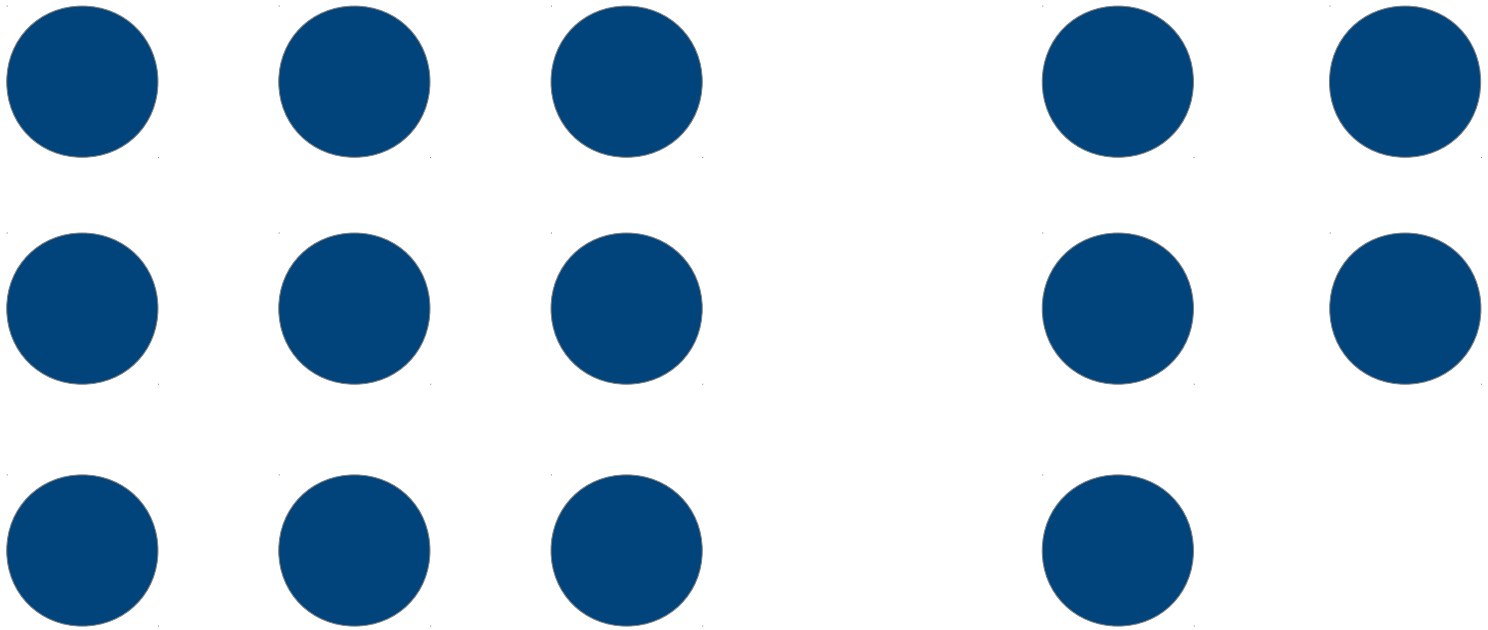


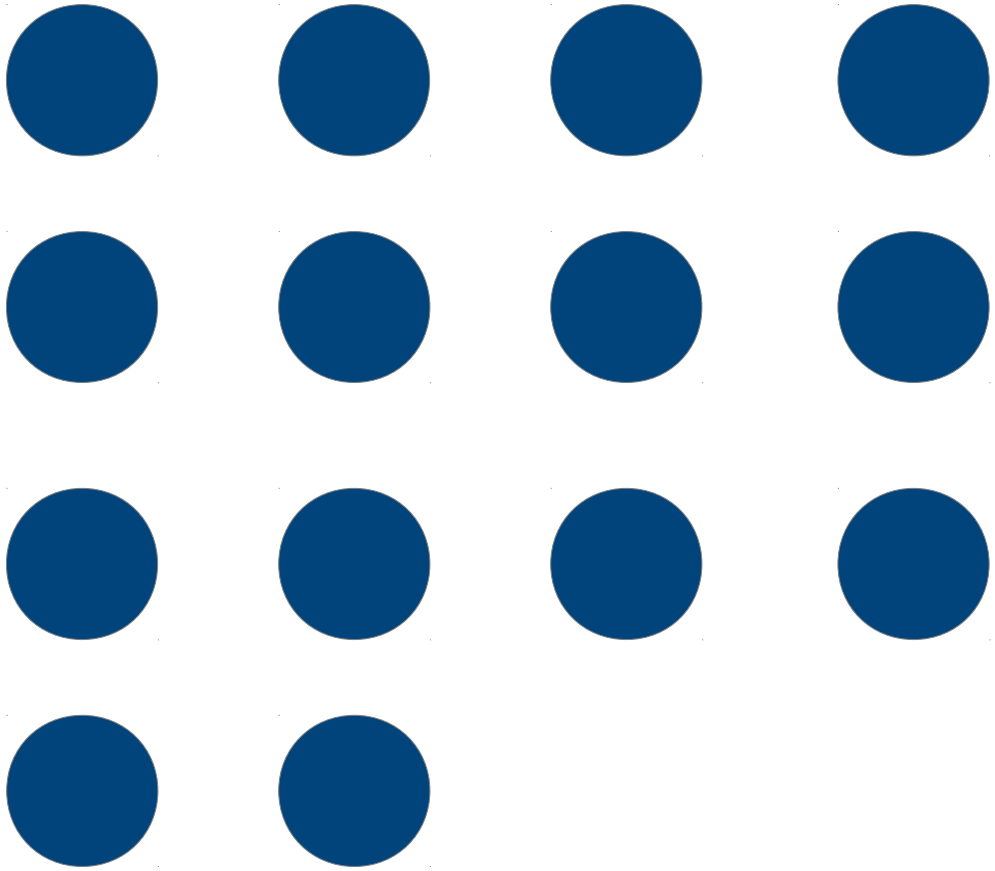












Sue runs around the track twice, and John runs around the track four times more than Sue.

How many times does John run around the track?

Sue runs around the track twice, and John runs around the track four times more than Sue.

How many times does John run around the track?

- $2 \times 4 = 8$

Sue runs around the track twice, and John runs around the track four times more than Sue.

How many times does John run around the track?

- $2 \times 4 = 8$
- $2 + 4 = 6$

Sue runs around the track twice, and John runs around the track four times more than Sue.

How many times does John run around the track?

- $2 \times 4 = 8$
- $2 + 4 = 6$
- $2 + (2 \times 4) = 10$

Beispiele

```
int i = 0;
if (x < y) {
    …
} else {
    …
}
f(i);
```

```
int i;
if (x < y) {
    …
} else {
    …
}
f(i);
```

```
int i = 0;
if (x < y) {
    i = x;
}
```

```
for (int i = 0, n = s.length(); i < n; i++) {
    …
}

int i = 0;
int n = s.length();
while (i < n) {
    …
}
```

```
if (3 == x) {
    …
}
```

```
if (“xxx”.equals(x)) {
    …
}
```

```
if (3 == x) {
    …
}
```

```
if (x == 3) {
    …
}
```

```
if (“xxx”.equals(x)) {
    …
}
```

```
if (x.equals(“xxx”)) {
    …
}
```

```
// Check to see if the employee is eligible for full benefit
if (
     (employee.flags & HOURLY_FLAG)
     &&
     (employee.age > 65)
) {

     …
}
```

```
if (employee.isEligibleForFullBenefits()) {
     …
}
```

```
f = open(…);

…

close(f);
```

```
f = open(…);

// Die Datei kann hier nicht geschlossen werden, da sie
// asynchron bearbeitet wird. Es obliegt damit dem Service
// für eine ordentliche Freigabe zu sorgen.

new Service(f).start();
```

```
/**
 * The month of the year.
 */
private int monthOfTheYear;
```

```
/**
 * The month of the year, expressed
 * as a value between 1 and 12.
 */
private int monthOfTheYear;
```

```
// Some day, we'll write a stopping test that takes
// account of the asymmetry of the spacing of floating-
// point numbers below perfect powers of 2.
// 26 Sept 96 is not that day.
// So we use a symmetric test.
```

Was macht dieser Code?


```
// Requires positive x
static int stringSize(int x) {
    for (int i = 0; ; i++)
        if (x <= sizeTable[i])
            return i + 1;
}
```

Was macht dieser Code?


```
final static int [ ] sizeTable = {
    9, 99, 999, 9999, 99999, 999999, 9999999,
    99999999, 999999999, Integer.MAX_VALUE
};

// Requires positive x
static int stringSize(int x) {
    for (int i = 0; ; i++)
        if (x <= sizeTable[i])
            return i + 1;
}
```

Was macht dieser Code?


```
// Fall thru to fast mode for smaller numbers
// assert i ≤ 65536;
for (;;) {
    q = (i * 52429) >>> (16 + 3);
    r = i - ((q << 3) + (q << 1));
    buf[--charPos] = digits[r];
    i = q;
    if (i == 0) break;
}
```

Was macht dieser Code?


```
// Fall thru to fast mode for smaller numbers
// assert i ≤ 65536;
for (;;) {
    q = (i * 52429) >>> (16 + 3);
    r = i - ((q << 3) + (q << 1));        // r = i - (q * 10)
    buf[--charPos] = digits[r];
    i = q;
    if (i == 0) break;
}
```

```
// Fall thru to fast mode for smaller numbers
// assert i ≤ 65536;
for (;;) {
    q = (i * 52429) >>> (16 + 3);      // q = i / 10
    r = i - ((q << 3) + (q << 1));     // r = i - (q * 10)
    buf[--charPos] = digits[r];
    i = q;
    if (i == 0) break;
}
```

- Alles hat eine Bedeutung
- Alles kann missverstanden werden

- Alles hat eine Bedeutung
- Alles kann missverstanden werden
- Kommentare und Dokumentation (und Unit-Tests) werden in der Regel Klarheit schaffen

- Ziele festlegen
- Zielpublikum festlegen

- Einem Schiff ohne Ziel ist kein Wind günstig
- (ignoranti quem portum petat nullus suus ventus est)
- Für den, der nicht weiß, welchen Hafen er anstrebt, ist kein Wind der seine
- Lucius Annaeus Seneca (der Jüngere) 1 – 65 n. Chr.

Always code as if the guy
who ends up maintaining your code
will be a violent psychopath
who knows where you live.

- Fragen?

- Vielen Dank!