



**BERGISCHE  
UNIVERSITÄT  
WUPPERTAL**

Prof. Dr. Hans-Jürgen Buhl  
Praktische Informatik/Numerik

Fakultät für  
Mathematik und Naturwissenschaften,  
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de

WWW www.math.uni-wuppertal.de/~buhl

DATUM 30. Oktober 2017

## **Softwarequalität**

**WS 2017/2018 – Übungsblatt 2**

**Ausgabe: 2. November 2017**

**Abgabe bis 8. November 2017 an: [1449250@uni-wuppertal.de](mailto:1449250@uni-wuppertal.de)**

### **Aufgabe 1. Programmverifikation**

Beschreiben Sie den Nutzen der Programmverifikation (Quelle: [http://en.wikipedia.org/wiki/Program\\_verification](http://en.wikipedia.org/wiki/Program_verification)). Warum setzt die Programmverifikation eine (formale) Programmspezifikation voraus? Erläutern Sie die Verifikation des GCD-Algorithmus

```
begin { $a > 0, b \geq 0$ }  
   $x := a; y := b;$   
  while  $y \neq 0$  do { $\text{gcd}(a,b) = \text{gcd}(x,y)$  }  
    begin  $r := x \bmod y;$   
       $x := y;$   
       $y := r$   
    end  
  { $x = \text{gcd}(a, b)$ }  
end
```

(entnommen: Suad Alagić/Michael A. Arbib: THE DESIGN OF WELL-STRUCTURED AND CORRECT PROGRAMS, Springer-Verlag, New York, 1978)

in eigenen Worten. „Reicht“ die Verifikation eines Programms als Qualitätssicherung aus?

### **Aufgabe 2. Anti-Pattern**

Lesen Sie den Artikel:

[http://en.wikipedia.org/wiki/Coding\\_by\\_exception](http://en.wikipedia.org/wiki/Coding_by_exception)

Beschreiben Sie in eigenen Worten (in deutscher Sprache) die dort geschilderten „Anti-Pattern“.

Was ist ein **Pattern**, was ein **Anti-Pattern**?

### **Aufgabe 3.** *Magische Zahlen*

Was versteht man beim Programmieren unter *magischen Zahlen* ([http://en.wikipedia.org/wiki/Magic\\_number\\_\(programming\)](http://en.wikipedia.org/wiki/Magic_number_(programming)))?

Warum sind sie schlecht für die Software-Güte? Wie sollten sie vermieden werden?

Welche Qualitätsanforderungen verletzt Code mit „magic Pushbuttons“ ([http://en.wikipedia.org/wiki/Magic\\_pushbutton](http://en.wikipedia.org/wiki/Magic_pushbutton))?

### **Aufgabe 4.** *Test-driven development und Unit-Tests*

Lesen Sie [http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development) und beschreiben Sie in eigenen Worten, wo und wie die Softwarespezifikation in der testgetriebenen Softwareentwicklung zum Tragen kommt.

Was sind Unit-Tests?

Welche Unit-Test-Werkzeuge stehen für C++ zur Verfügung ([http://en.wikipedia.org/wiki/List\\_of\\_unit\\_testing\\_frameworks#C.2B.2B](http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks#C.2B.2B))?

### **Aufgabe 5.** *freie Entwickler-Tools zur Sicherung hoher Softwarequalität*

Durchsuchen Sie die auf der Seite

<http://packages.debian.org/de/sid/powerpc/devel/>

genannten Softwarepakete auf solche hin, die die Softwarequalität beim Entwicklungsprozess steigern können.

Erstellen Sie eine Liste dieser Pakete.