



Softwaregüte

WS 2012/13 – Übungsblatt 2

Ausgabe: 7. November 2012

Abgabe: bis spätestens 14. November 2012 an
Christian.Seepold@uni-wuppertal.de

Aufgabe 1. *Programmverifikation*

Beschreiben Sie den Nutzen der Programmverifikation (Quelle: 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. *Software-Güte*

Lesen Sie den Artikel:

http://en.wikipedia.org/wiki/Coding_by_exception

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

Wie sind sie in Bezug auf produkt- und projektorientierte Qualitätsanforderungen an Software einzuordnen?

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)?

Aufgabe 4. *Test-driven development*

Lesen Sie 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)?