Methodische Grundlagen des Software Engineering - Übung 9

9 Prozess und Softwarequalität

Abgabe der Hausaufgaben am Anfang der jeweiligen Präsenzübung am 14.06.2011 bzw. 15.06.2011.

9.1 Grundlagen Prozessmanagement

9.1.1 Was versteht man unter der ISO9001 und CMMI?

9.1.2 Wo überschneiden sich ISO9001 und CMMI und wo liegen Unterschiede? Können sich die beiden Standards ergänzen?

9.1.3 Welche der beiden folgenden Aussagen würdest du der ISO 9001 und welche CMMI zuordnen?
1) "Eignet sich sehr gut um die bestehend Qualitätsorientierung nach außen zu transparent zu machen und vertrauen zu schaffen"
2) "Hilft eine Qualitätsorientierung aufzubauen und zu verbessern"

9.2 CMMI

9.2.1 Gegeben folgendes Fähigkeitsprofil. Welche Reifestufe wurde von dem geprüften Unternehmen erreicht?

Reifegrad (gestufte Darstellung)	Prozessgruppe (Verlaufsdarstellung)			
	Projekt-management (project management)	Unterstützung (support)	Technik (engineering)	Prozess-management (process management)
5 Optimierend (optimizing)		CAR		OID
4 Quantitativ geführt	QPM			OPP
3 Definiert (defined)	IPM RSKM	DAR	PI RD TS VAL VER	OPD OPF OT
2 Geführt (managed)	PMC PP SAM	CM MA PPQA	REQM	

9.2.2 Welche Prozessgruppen müssten um eine Fähigkeitsstufe angehoben werden um eine höhere Reifestufe zu erreichen?

9.2.3 Welche Reifestufe wird dadurch erreicht?

9.2.4 Welches minimale Fähigkeitsprofil ergibt sich wenn Reifestufe 3 erreicht wurde?

9.2.5 Ist der parallele Einsatz von Verlaufs- und gestufter Darstellung innerhalb eines Unternehmens sinnvoll?

9.3 Messtheorie

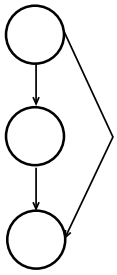
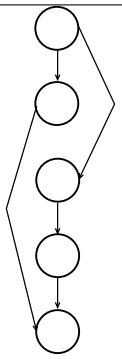
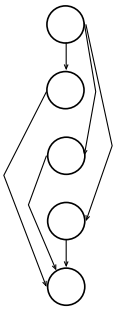
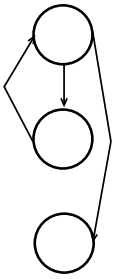
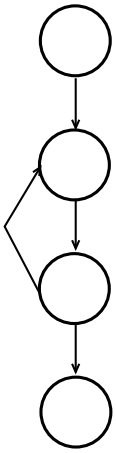
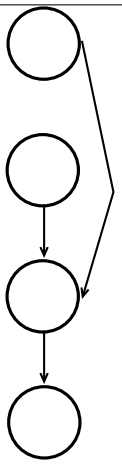
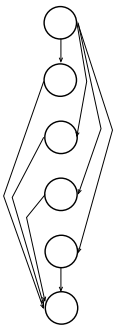
9.3.1 Woraus besteht ein Maß?

9.3.2 Welche Skalentypen gibt es? Gib jeweils ein Beispiel.

9.3.3 Wie kann man 2 unterschiedliche Maße vereinheitlichen? Das heißt was kann man tun um 2 Mengen von gleichartigen Objekten, die aber mit unterschiedlichen Maßen bewertet wurden, vergleichbar zu machen? Wo liegen dabei Probleme?

9.4 Zyklomatische Komplexität

9.4.1 Gib jeweils die zyklomatische Komplexität für folgende Konstrukte an.

Kontrollflussgraph				
Typ	If Anweisung	If else Anweisung	Switch Anweisung (2 x case/break 1xdefault)	While Anweisung
zyklomatische Komplexität				
Kontrollflussgraph				
Typ	do while	goto	Switch Anweisung (3 x case/break 1x default)	
zyklomatische Komplexität				

9.4.2 Welche Erkenntnisse folgerst du aus den Ergebnissen aus 9.4.1?

9.4.3 Zeichne den Kontrollflussgraph zu folgender Funktion und gibt die zyklomatische Komplexität an.

```
void output( int selection, int y){
        if (selection ==1){
                printf("\n 1");
        }
        else {
                if (selection ==2) {
                        printf("\n 2");
                }
                else{
                        if (selection ==3){
                                printf("\n 3");
                        }
                        else{
                                if (selection ==4){
                                        if (y == 0){
                                                printf("\n 4");
                                                printf("\n 5");
                                        }
                                        else{
                                                printf("\n 6");
                                        }
                                }
                                else{
                                        printf("\n 7");
                                }
                        }
                }
        }
}
```

9.5 Halstead Metriken

9.5.1 Fülle die gegebenen Tabellen aus und gib die Halstead Metriken N,L,V,E und T für folgendes PASCAL Programm an.

Hinweis: E war in den Folien falsch definiert. Dies wurde am 11.06 korrigiert und die neuen Folien online gestellt. Definition von Operatoren und Operanden:

- Es werden nur Statements im Ausführungsteil betrachtet.

- Operatoren sind:
  - arithmetische Operatoren: +, -, *, /, DIV, MOD
  - boolesche Operatoren: NOT, AND, OR
  - relationale Operatoren: <, >, <=, >=, =, <>
  - Mengenoperatoren: IN
  - String-Operatoren: +
  - Zuweisungen: :=
  - Schlusselwörter: BEGIN END, IF THEN, ELSE, WHILE DO, FOR DO
  - spezielle Operatoren:
    - Funktionsaufruf
    - Klammern: ( )
    - ARRAY-Klammern: [ ]
    - Statement-Begrenzungszeichen: ;
    - Programm-Begrenzungszeichen: .
    - Unterprogramm-Begrenzungszeichen: ;
    - GOTO
- Operanden sind:
  - Variablennamen
  - Konstantennamen (auch Fixwerte)
  - Funktionsnamen
  - Sprungmarken

```
FUNCTION PrimTest(input: INTEGER): BOOLEAN;
VAR n, teiler: INTEGER;
 test: BOOLEAN;
BEGIN
        n := input;
        IF n >= 2 THEN
                BEGIN
                        teiler := 2;
                        WHILE (teiler < n) AND ((n MOD teiler) <> 0) DO
                        teiler := teiler +1;
                        IF teiler = n
                        THEN test := true
                        ELSE test := false;
                END
        ELSE test := true;
```

```
        PrimTest := test;
END;
```

Operator	Häufigkeit	Operand	Häufigkeit
$n_1 =$	$N_1 =$	$n_2 =$	$N_2 =$

Hausaufgabe

9.6 Zyklomatische Komplexität

9.6.1 Zeichne den Kontrollflussgraph zu folgender Funktion und gibt die zyklomatische Komplexität an.

```
int wandleDezimalZahl() {
        int Dezimalzahl = 0;
        int Potenzwert = 0;
        char Zchn;
        Zchn = leseNaechsteZeichen();
        while (((Zchn == "0") || (Zchn == "1")) && (Dezimalzahl < INT_MAX)){
                if (Zchn == "1") {
                        Dezimalzahl = Dezimalzahl + Math.pow (2, Potenzwert);
                }
                Potenzwert = Potenzwert + 1;
                Zchn = leseNaechsteZeichen();
        }
        return Dezimalzahl;
}
```

2 P

9.7 Halstead Metriken

9.7.1 Fülle die gegebenen Tabellen aus und gib die Halstead Metriken N,L,V,E und T für folgendes PASCAL Programm an.

Hinweis: E war in den Folien falsch definiert. Dies wurde am 11.06 korrigiert und die neuen Folien online gestellt. Definition von Operatoren und Operanden:

- Es werden nur Statements im Ausführungsteil betrachtet.
- Operatoren sind:
  - arithmetische Operatoren: +, -, *, /, DIV, MOD
  - boolesche Operatoren: NOT, AND, OR
  - relationale Operatoren: <, >, <=, >=, =, <>
  - Mengenoperatoren: IN
  - String-Operatoren: +
  - Zuweisungen: :=
  - Schlusselwörter: BEGIN END, IF THEN, ELSE, WHILE DO, FOR DO

  - spezielle Operatoren:
    * Funktionsaufruf
    * Klammern: ( )
    * ARRAY-Klammern: [ ]
    * Statement-Begrenzungszeichen: ;
    * Programm-Begrenzungszeichen: .
    * Unterprogramm-Begrenzungszeichen: ;
    * GOTO

- Operanden sind:
  - Variablennamen
  - Konstantennamen (auch Fixwerte)
  - Funktionsnamen
  - Sprungmarken

```
FUNCTION PrimTest(input: INTEGER): BOOLEAN;
VAR stop, teiler: INTEGER;
 test: BOOLEAN;
BEGIN
        teiler := 2;
        stop := TRUNC(SQRT(input));
        test := true;
        WHILE (teiler <= stop) AND test DO
                BEGIN
                        test := input MOD teiler <> 0;
                        teiler = teiler + 1;
                END;
        PrimTest := test;
END;
```

Operator	Häufigkeit	Operand	Häufigkeit
$n_1 =$	$N_1 =$	$n_2 =$	$N_2 =$

3 P