Student ID: **ID-No.:**

Systems and Software Engineering
Examination ST 2013

Institute for Information Processing Technologies– ITIV
Prof. Dr.-Ing. Müller-Glaser

Systems and Software Engineering
Date: 13.09.2013
Name:
Matriculation ID:

Lecture Hall:
Seat No.:

Prerequisites for the examination

Aids

- Allowed aids for the examination are writing utensils and a single sheet of A4 paper with self- and hand-written notes. Writing may be on both sides of the paper. The use of own concept paper is not allowed.
- Use only indelible ink - use of pencils and red ink is prohibited.
- Other material than that mentioned above, is strictly forbidden. This includes any type of communication to other people.

Duration of the examination

120 minutes

Examination documents

The examination comprises 22 pages (including title page). Answers may be given in English or German. A mix of language within a single (sub)-task is not allowed. In your solution mark clearly which part of the task you are solving. Do not write on the backside of the solution sheets. If additional paper is needed ask the examination supervisor.
You will not be allowed to hand in your examination and leave the lecture hall in the last 30 minutes of the examination.
At the end of the examination: Stay at your seat and put all sheets into the envelope. Only sheets in the envelope will be corrected. We will collect the examination.

		Page	Points	Result
Task 1	General Questions	2	13	
Task 2	Quality function deployment	5	7	
Task 3	Petri Nets	7	16	
Task 4	State Charts	10	10	
Task 5	Scheduling	11	13	
Task 6	Reliability	14	10	
Task 7	UML Diagrams	16	19	
Task 8	Other diagrams	20	4	
			92	

1 General Questions

1.a Software test

/2

Name the different phases of the software test process, discussed in the lecture.

.....

.....

.....

.....

.....

/1

1.b BNF

Explain shortly the reason for using the Backus-Naur-Form.

.....

.....

/3

1.c UML

Name eight different UML 2.0 diagrams and classify them in Table 1.

Structural UML diagrams	Behavioral UML diagrams

Table 1: UML diagrams overview

/2

1.d Hunger Life Cycle Model

What is the difference between the Sortie Mission and the Life Mission?

/2

1.e V-model

Name the four sub models of the V-Model ‘97.

Multiple choice

You get plus 0.5 points for a right answer and minus 0.5 points for a wrong answer. In total you cannot get a negative number of points for each subtask. More than one answer could be true.

1.f Real time systems

/1.5

Select the correct answers:

true	false	
☐	☐	Hard real-time systems need faster hardware than soft real-time systems.
☐	☐	The response time of a computing process in a real-time system is the sum of waiting-, input-, processing- and output-time.
☐	☐	In the context of a real-time operating system, the “first come first served” - strategy minimizes the average response time when tasks arrive at the same time.

1.g Watchdog timer

/1.5

Which statements could be made for software watchdog timer and hardware watchdog timer?

true	false	
☐	☐	It is not possible to implement both, hardware and software timer in one system.
☐	☐	Timer in software runs much slower than the processor clock.
☐	☐	Time resolution of the hardware timer has in best case the same resolution as the system clock.

2 Quality function deployment

2.a House of Quality

/2

What is the primary function of the House of Quality tool?

.....

.....

.....

.....

2.b House of Quality

/2

Draw the shape of a House of Quality with all its components.

/2

2.c House of Quality

Name all elements in the House of Quality and point out the location in the figure you drew in 2.b.

/1

2.d House of Quality

Which gradations are commonly used in the relationship entries?

3 Petri nets

/4

3.a Petri net characteristics

Which of the following characteristics (a-d) of petri nets are mutually exclusive? Give a short explanation for each.

a) Alive and reversible

b) Alive and dead marking

c) Reversible and not alive

d) Reversible and not safe

a) Mutually exclusive? Yes ☐ No ☐

Explanation:

.....

.....

.....

b) Mutually exclusive? Yes ☐ No ☐

Explanation:

.....

.....

.....

c) Mutually exclusive? Yes ☐ No ☐

Explanation:

.....

.....

.....

d) Mutually exclusive? Yes ☐ No ☐

Explanation:

.....

.....

.....

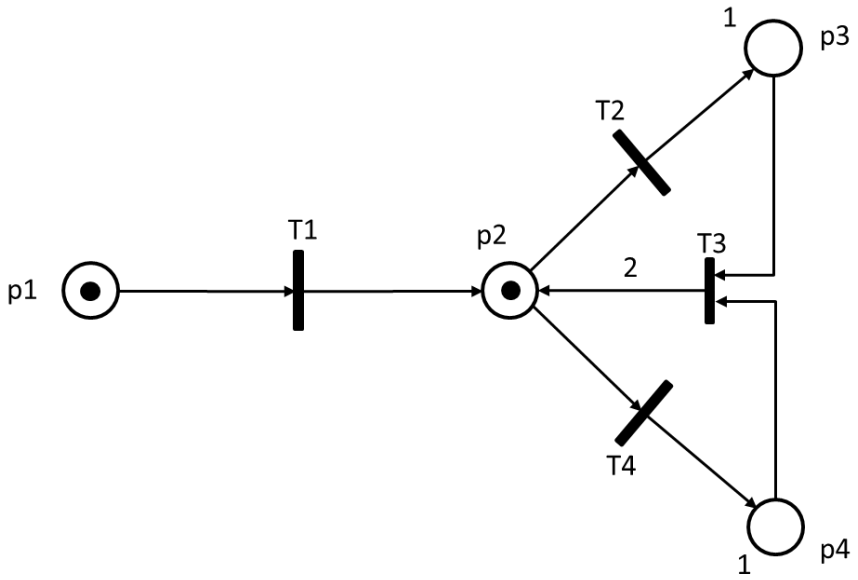

Figure 1: Petri net

/5

3.b Petri net characteristics

Is the petri net shown in Figure 1 safe? Draw the transition graph and explain your answer.

.....

.....

.....

/2

3.c Petri net characteristics

Is the petri net shown in Figure 1 alive? Explain your answer.

/5

3.d Design a petri net

Consider a circular railroad system with 4 tracks and 2 trains (represented by simple tokens). No two trains should be on the same track at the same time. The 4th track is connected to the 1st track. Between two tracks there is a station located.

Model a Petri net with simple tokens, so that both trains can use the tracks and travel conflict-free through the entire railroad system.

Point out the capacity of each place and the initial tokens to start from. Use *station 1* to *4* and *track 1* to *4* as labels for the specific parts of the net.

4 State charts

/2

4.a History connector

How is history connector “H” treated in a state chart diagram?

.....

.....

.....

4.b Design a state chart

/8

Design a state chart for a simple coffee dispensing machine. The machine has two phases: in a first phase the money for the desired drink is collected. The money that can be inserted is 20 cent coins only. Each beverage costs 40 cents. During the collection phase money can always be reclaimed. If at least 40 cents are stored in the machine, the user can choose between coffee, decaffeinated coffee and tea. After the drink was prepared the machine outputs the cup and returns to the collection phase. Pay attention to the following malfunction of the machine which may occur at any time: an error could occur and prevent the machine from behaving correctly. After such a malfunction is repaired, the machine automatically returns to the collecting phase.

5 Scheduling

/8

5.a Task scheduling

Four tasks with different priority should be executed on one processor. The following table shows the features of these tasks.

Task	Processing Time	Priority (0 is highest)	Arrival time	Deadline
A	32	2	T + 0 ms	T + 100 ms
B	19	0	T + 7 ms	T + 125 ms
C	45	1	T + 14 ms	T + 90 ms
D	27	3	T + 19 ms	T + 110 ms

Table 2: Scheduling

You are at a certain point in time T, all tasks request processor-time from this point in time. Plot the processing of the given tasks into the following diagrams (see next page Figure 2) under consideration of the scheduling methods named below.

- Round Robin (time slice 10ms) (Task Queue)
- TDMA (time slice 15ms) (Cycle A-B-C-D, A starts at T)
- Priority Scheduling
- Deadline Scheduling

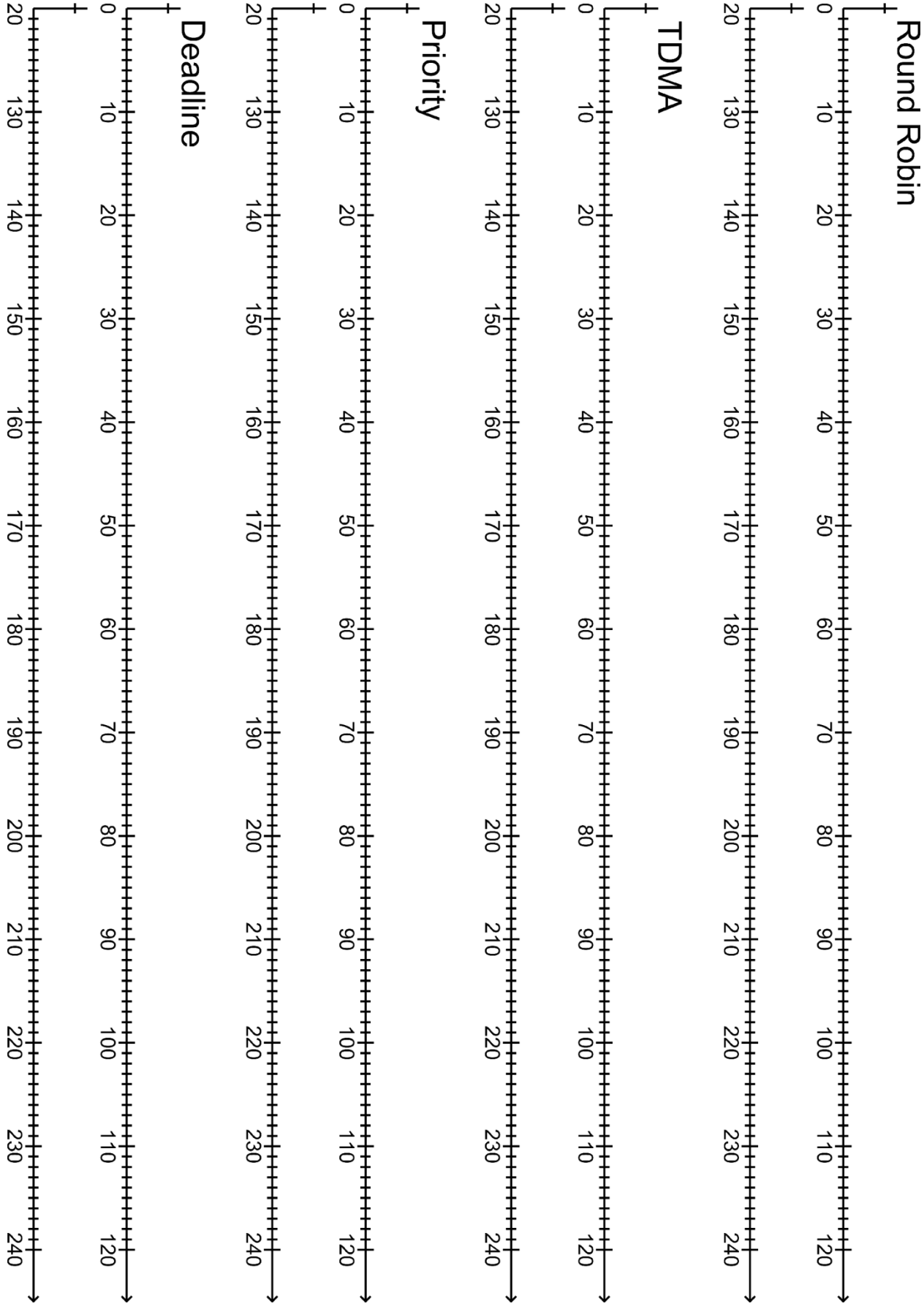

Figure 2: Scheduling

/5

5.b Response time

Calculate the maximal and the average response time of every scheduling method described in 5.a.

6 Reliability

Figure 3 shows part of a switchgear. The shown setup allows switching from the main power transformer T1 to the reserve power transformer T2.

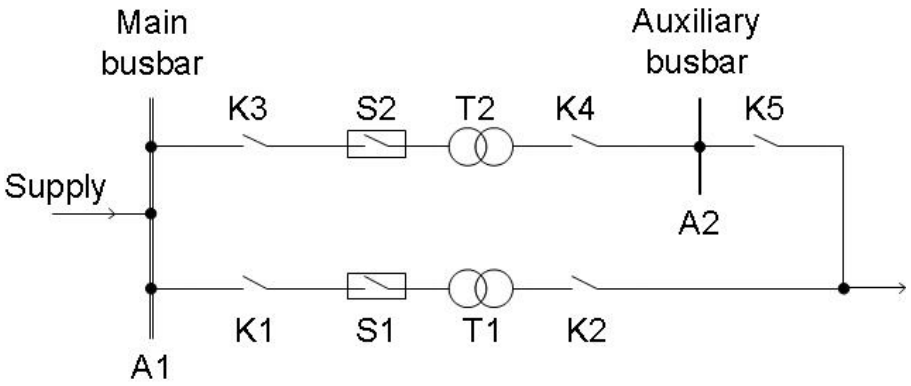

Figure 3: Part of a switchgear

Component type	Amount	Code	λ [10^{-6} / h]
Busbar	2	A1, A2	1,0
Power transformer	2	T1, T2	5,0
Disconnector	5	K1-K5	3,0
Power switch	2	S1, S2	10,0

Table 3: Switchgear components and failure rates

/1

6.a MTTF

Calculate the MTTF for a single power transformer.

/3

6.b Reliability block diagram

Draw a reliability block diagram for the whole system shown in Figure 3.

/6

6.c Reliability and MTTF

Determine the reliability R(t) and the MTTF of the electronic switch shown in Figure 3. The answer should compromise the intermediate steps of your calculation.

7 UML diagrams

```
class A {
protected:
  int value;
public:
  void C1()  {
    cout << "C1()" << endl;
  }
};
class B : public A {
public:
  void C2()  {
    cout << "C2()" << endl;
  }
  int calc() {
    return value * 10;
  }
};
class C : public A {
public:
  void C3()  {
    cout << "C3()" << endl;
  }
};
class D : public B {
private:
  C c2;
public:
  void C4()  {
    cout << "C4()" << endl;
  }
};

int main () {
  D d;
  return 0;
}
```

Figure 4: C++ software code

7.a Class diagram

/5

Draw the corresponding class diagram of the C++ software code given in Figure 4. The class diagram should include as many information as possible like classes, attributes, operations and associations.

7.b Object diagram

/4

Draw the corresponding object diagram of the C++ software code given in Figure 4 with all possible information.

/6

7.c UML activity diagram

a) Analyze the program given below in Figure 5 by drawing an UML activity diagram.

b) Which task does the function *func()* fulfill?

```
void func (int x[], int y)  {
  for (int i=0; i<y-1; i++)  {
    for (int j=0; j<y-i-1; j++)  {
      if (x[j] > x[j+1])  {
        int z = x[j];
        x[j] = x[j+1];
        x[j+1] = z;
      }
    }
  }
}
```

Figure 5: C++ software code

b)

7.d UML use case diagram

/4

Develop a use case diagram based on the following description of visiting a party.

“A child can visit a party. Eating and drinking soft drinks are the things a child does on a party. On a party, an adult can do everything a child can. In addition to a child, the adult can drink alcohol. Drinking alcohol may result in getting drunk, but it doesn’t need to.”

8 Nassi Shneiderman diagrams

/4

8.a Nassi Shneiderman diagram

Build a Nassi-Shneiderman diagram from the pseudo code program stated below.

```
float doSomething(float x, int r) {
  if (x == 0) return 1;
  float v = 1;
  float w = 1;

  x *= -x;
  if (r<1) {
    r=10;
  }
  int i=1;
  do{

    w = w*x;

    w = w/(2*i-1);

    w = w/(2*i);
    v += w;
    ++i;
  } while ( i<=r );
  return v;
}
```

Additional working sheet:

Exercise number: ____

Additional working sheet:

Exercise number: ____