ICT Service Management
Example Questions

1. Consider the following product description for a software product, called ZBB. ZBB is using a three-tier client-server processing approach, using client workstations, one or more application servers and one single database server. In the initial version of ZBB the database server is based on the ORACLE DBMS, the application server is written in C++ (for performance reasons) whereas the client is based on UNIX graphical user interfaces, using a GUI-builder called SuperGUI. Client, application and database servers are supposed to run SUN SOLARIS, the SUN version of the UNIX operating system.

   The software group that developed ZBB needed 3 years (and 3 releases) with 25 people to arrive at a stable version. SUPERGUI will generate JAVA, running on all platforms under any WWW-browser, in 1 year from now. At the same time there is a strong demand for support for the IBM AIX operating system (the IBM version of the UNIX operating system), on the application and database servers. A non-neglictable group asks support for Windows/NT in 1 year from now, whereas it is seen that LINUX (the open source code version of UNIX) will become a significant operating system for workstations/clients as well as servers. ORACLE runs on all the platforms that are mentioned. IBM wants to invest in additionally 10 people during two years if the product is migrated to DB2, IBM's alternative to ORACLE. Needless to mention that some Windows/NT customers are interested in an SQL/Server (the Microsoft alternative for ORACLE) for the database server. It should be realized that all database systems involve different C++-code to use them.

   Suppose you are the product manager for ZBB for the coming 4 years. Develop a branching and development strategy for ZBB for the next 4 years, based on the above information. You may count on a global market expansion of 20% on a yearly basis, and the company producing ZBB was so far break-even on the above number of developers !

2. The configuration manager, aware of a potential multiple maintenance problem, knows that it is foolish to try to keep seven copies of the same COMMON declaration in synchronization in a shared program environment. So the baseline repository stores only one copy of that COMMON declaration, and a macro facility (INCLUDE) is used to insert the COMMON declaration into each module as part of the compilation process.

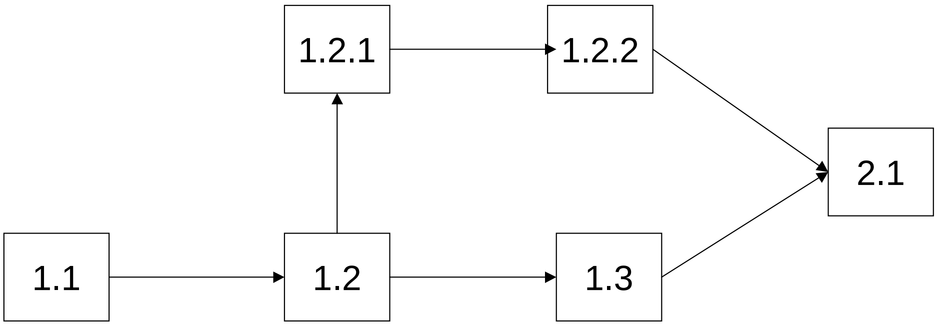
   Is this a safe procedure ? Find potential problems and propose alternatives ?

3. *Mark's problem is solved. Though it takes a week to make the bug reproducible and another two weeks to find the bug and fix it, Mark was able to proceed in complete stability, independent of the other project members.*

Mark finds the bug and charges out the modules that need to be fixed. Then he conducts rigorous tests to make sure that the problem is really gone. When his new system survives without crashing for an entire week, despite his best efforts to make it fail, Mark concludes that the bug is really fixed. He charges the fixed module into the baseline.
Mark barely has time to plan the celebration before the other project members come banging on his door. Everything stopped working. The whole world had falled apart.

What happened here ? How would you prevent the above problem ?

4. What are the necessary steps (formulate the deltas) to realize the following product plan ?



5. Try to interprete the following make-file and give a description of the product space

```
runme    :    a.o     b.o    c.o
              linkit  a.o    b.o    c.o    runme
a.o      :    a.c     incl.h
              compile        a.c
b.o      :    b.c     incl.h
              compile        b.c
c.o      :    c.c
              compile        c.c
```

6. In a particular real-time programming project, all the source and object code for the multiple modules is stored on an online disk which is shared among the project members.
Since there is only one copy of the source code there is no danger for a double maintenance problem…
The software development is quite far along, with a larger portion of the program integrated and working. Bill Barnstormer, crack software engineer, has a few bugs left in his input routines. Bill uses an editor to repair the bugs by editing the online source files. Then he recompiles the affected modules, creating new object code that overwrites the old ones…

a. ***Can Bill do this without accidentally introducing a new bug ?***
b. ***If Bill's fixes are defective, in which module will the new bug manifest itself:***
   - ***the input routines that Bill modified, or***
   - ***some other module that someone else is working one ?***