PROBLEM 1: CREATE A CAUSAL LOOP DIAGRAM.

In this problem, you will create a causal loop diagram for dynamic behavior associated with some situation you have encountered in your life or with which you are familiar.[1] The diagram must include at least 5 factors (labels serving as nodes in the directed graph), and there must be at least 3 different possible paths from start to finish. You are not required to be absolutely sure about the real-world causal relationships to model such a situation with a causal loop diagram; you can create a causal loop diagram as a working hypothesis for the pattern behind a situation, even if you don't know (or can't recall) all of the detailed mechanisms. Greater credit will be awarded for causal loop diagrams that exhibit greater initiative, clarity of thought, and creativity in conceptualization. While you will be handing in only one diagram for this problem, it is suggested that you follow the following procedure:

a) Think about what causal factors might play a role in the behavior you have seen, and write them down in a list

b) You may draw the causal loop diagram by hand or in a drawing package (a drawing package can add burden to drawing things out, but packages supporting "sticky" connections can compensate by easing rearrangement of the diagram). Lay out the factors in such a way that you could easily draw connections between those likely to directly influence one another. Based on this layout, draw the directed edges (arrows) between factors. Each such edge should ideally represent some specific causal pathway (a particular influence, or a tight cluster of types of influences). Typically this is a somewhat iterative process, where you create the labels for the various factors, start thinking through the causal influences (which you can sketch in with directional edges – i.e. arrows), add or refine factors, rearrange, etc. Try to capture in the diagram the set of all effects that you believe have a substantial impact on the behavior or phenomenon to be explained. Be patient and you will likely come up with a clearer depiction of the situation than you would have in a first, rushed attempt.

c) For each edge (arrow), indicate the *polarity* of the influence, by asking yourself whether an increase in the source factor would lead to an increase or decrease in the destination factor. If the polarity is ambiguous, you should probably go back to the previous step and add in additional causal pathways (links with particular causal meanings) between the source and destination, each of which should have some unambiguous polarity.

d) Once you have converged on the *link* polarities, it is time to label the *loop* polarities – at least for the major loops[2]. For the most obvious/major loops, draw a partial circle (with an arrow in the direction of the loop) with a "+" inside for a positive (reinforcing) feedback loop, and a "-" inside for a negative (balancing) feedback loop (e.g.).

[1] Note that the dynamics and causal loop diagram that results should be different from those you work on in problem 2.

[2] Note that the combinatorics is such that each time we add a cycle to the causal loop diagram, we likely add multiple cyclic pathways through the entire causal loop diagram. While there are ways of identifying a minimal size basis set of "independent" loops within a graph, the techniques are too involved to cover for our brief exposure to causal loop diagramming.

PROBLEM 2: VICIOUS CYCLES AND LOCK-IN EFFECTS.

In this problem, you will create causal loop diagrams to illustrate the effects of “vicious cycles” and lock-in effects that we mentioned in lecture.

a) Use a causal loop diagram to illustrate a “vicious cycle” involving developer morale, workload, and resignations. There are a number of effects that a diagram of this sort could include, and there is no canonical, completely “correct” depiction. But do try to come up with a diagram that illustrates at least a few pathways could plausibly affect these factors.

b) Using a causal loop diagram, give a plausible theory (or set of explanatory theories) as to how some projects might be quite successful both at managing risks and maintaining a more sustainable level of effort and higher morale, while others get “stuck” in a pattern in which they seem to be constantly in overload mode, “firefighting” (crisis-driven handling of unexpected events), fatigued and with low morale.

PROBLEM 3: QUALITY ATTRIBUTE TRADEOFFS.

In lecture, we briefly discussed some of the attributes we cluster under the term "quality". As discussed, many of these have been formally defined as "quality attributes" by different bodies (e.g. IBM, HP, ISO, etc.) For example, the ISO definition of quality attributes include the following:

- Functionality (Suitability,Accuracy,Interoperability,Compliance,Security)
- Reliability (Maturity, Recoverability, Fault Tolerance)
- Usability (Learnability, Understandability, Operability)
- Efficiency (Time & Resource Behavior)
- Maintainability (Stability, Analysability, Changeability, Testability)
- Portability (Installability, Conformance, Replaceability, Adaptability)

In this problem, you will be asked to reason about some of the tradeoffs associated with meeting quality attributes.

a) Briefly describe how *opportunity costs* of effort could cause tension in realizing different quality attributes within realistic project constraints.

b) As was briefly mentioned in class, even ignoring the reality of opportunity costs, there are tensions associated with realizing different attributes. Many of these tensions reflect technological constraints and tradeoffs – including many techniques you would have discussed in other classes. List ten (10) technological mechanisms/techniques that could cause tradeoffs between attributes (for example, particular technologies or approaches that might be undertaken to address one issue but which could hinder quality in one or more other areas), and briefly discuss what attributes might be affected by using this technique (or could motivate its use) and why. Please try to be brief but specific and clear as to why a given technique/technology/approach might affect a given quality attribute in a positive or negative way. To start you off, consider the tradeoffs associated with mechanisms to allow a system to be transactional. You may consider this as one of your ten mechanisms.

CMPT 816 Problem Set 1

Svetlana Slavova
sds797@mail.usask.ca

Problem 1: Create a Casual Loop Diagram

My life as a student at an abstract level

1) The main factors that play a role in my life as a student:
   - Work time spent at school (referred as time spent at work);
   - Social activities;
   - Happiness;
   - Stress level;
   - Tasks completion.

2) CLD diagram showing my student life at an abstract level

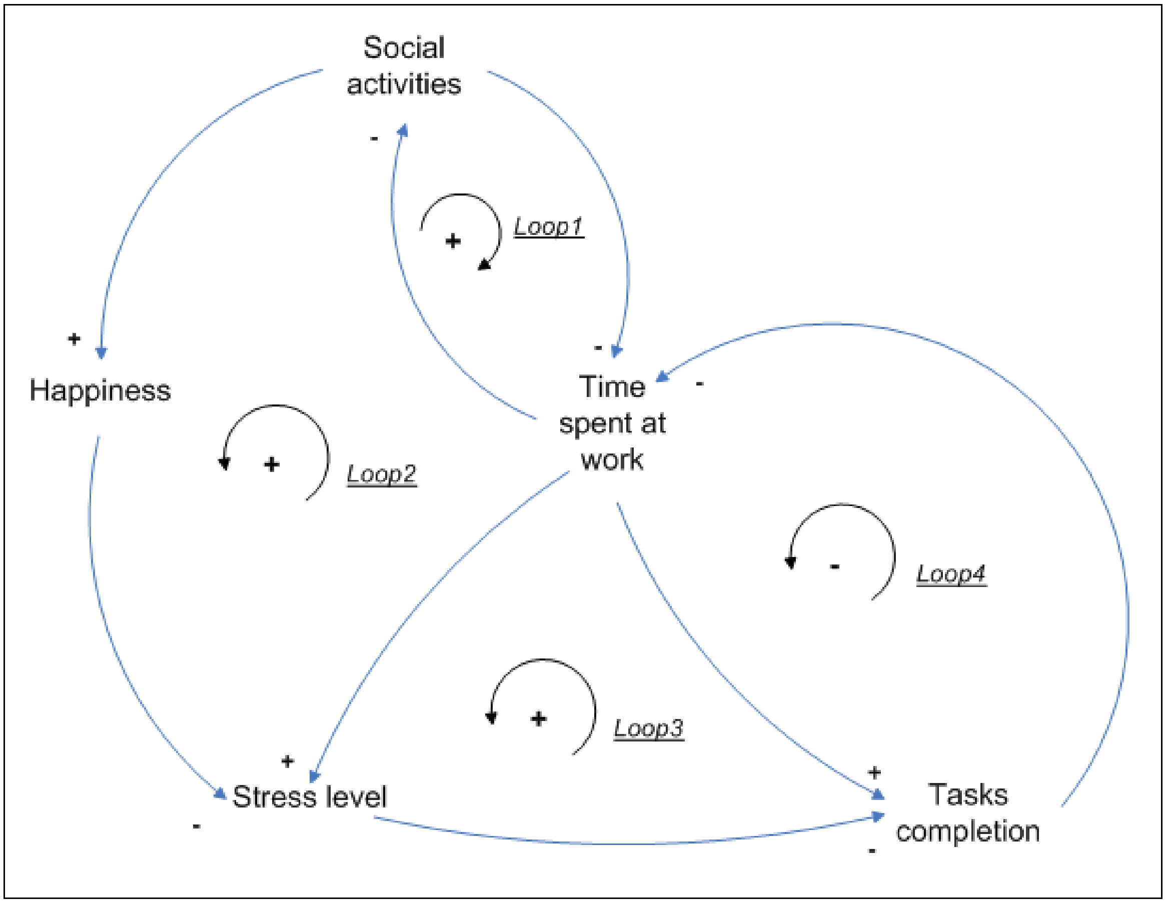


Figure 1. CLD representing my life as a student at an abstract level

CMPT 816 Problem Set 1

Svetlana Slavova
sds797@mail.usask.ca

The casual loop diagram (Figure 1) shows my life as a student at an abstract level, depending on the time spent at work.
There are four loops presented in the diagram – *Loop1* to *Loop4*; three of them are positive, reinforcing feedback loops, and one is a negative, balancing loop. The feedback loops are as follows:

- *Loop1 (+):* **Time spent at work à Social activities à Time spent at work**
  This is a positive feedback loop. When the time spent at work increases, it reflects the social activities which decreases and the time spent at work increases. The opposite is true as well – if less time is spent at work, there will be more social activities. When more social activities are done, then less time is spent at work.

- *Loop2 (+):* **Time spent at work à Social activities à Happiness à Stress level à Tasks completion à Time spent at work**
  This is another positive feedback loop. It shows that when the time spent at work increases, less social activities can be done. It reflects the happiness and the happiness decreases. Then the stress level increases which decreases the number of competed tasks. When the number of completed tasks decreases, then more time should be spent at work.
  The opposite is true as well – When less time is spent at work, there are more chances for social activities which increase the happiness. The increased level of happiness reduces the stress level and more tasks are completed. When more tasks are completed, less time can be spent at work.

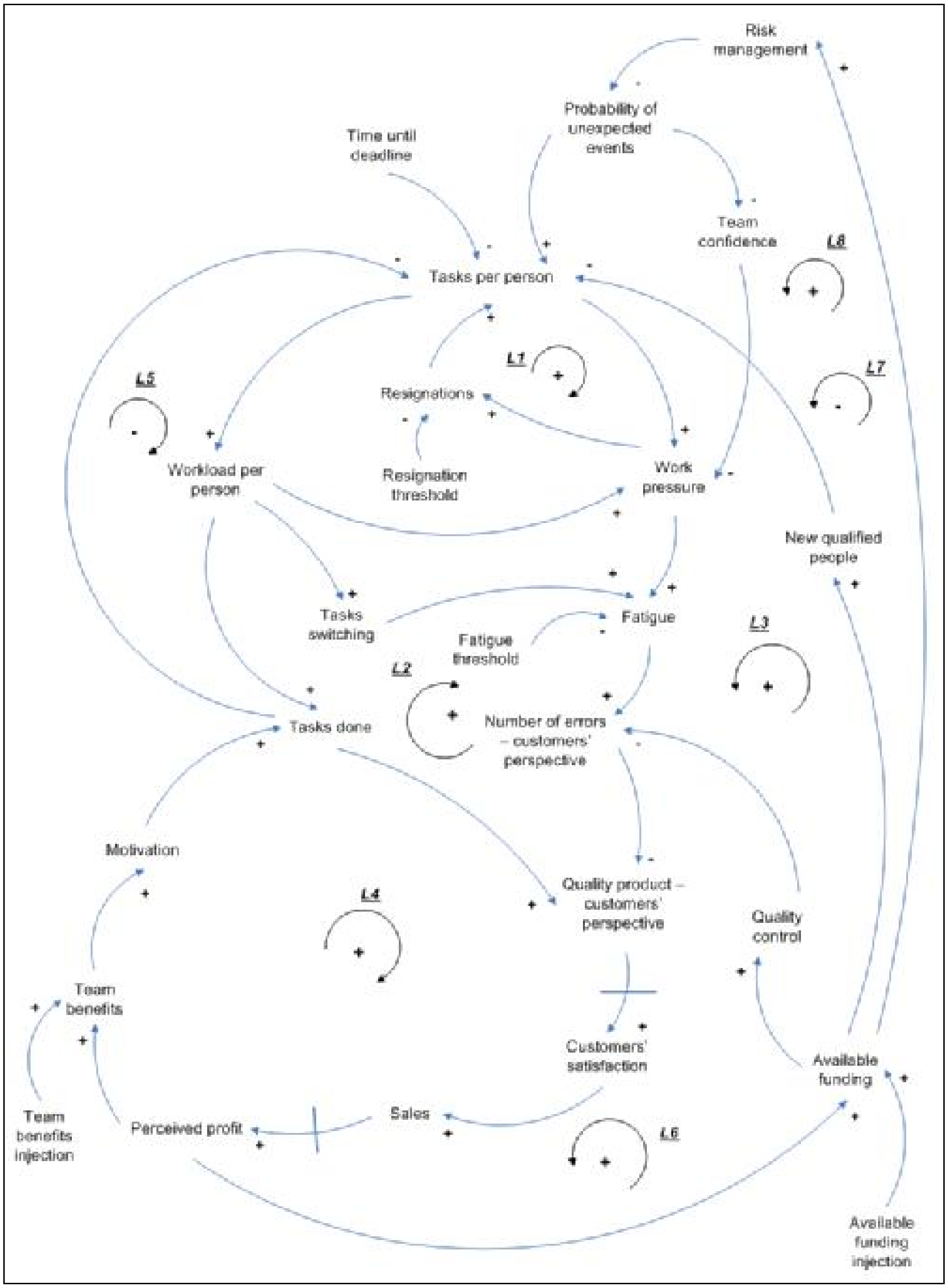
- *Loop3 (+):* **Time spent at work à Stress level à Tasks completion à Time spent at work**
  This positive feedback loop shows that when the time spent at work increases, the stress level increases too. The tasks completion decreases due to the increased stress level. This leads to the fact that more time has to be spent at work.
  On the other hand, when less time is spent at work, then the stress level is lower, and more tasks can be done efficiently. Then, less time can be spent at work.

- *Loop4 (-):* **Time spent at work à Tasks completion à Time spent at work**
  This loop is a negative, balancing loop. It shows that if the time spent at work grows up, the tasks completion increases as well. Then, there is no sense of spending so much time at work, so the time at work decreases.
  However, when the work time reduces, the tasks completion reduces as well. This requires more time at work to be spent.

CMPT 816 Problem Set 1

Svetlana Slavova
sds797@mail.usask.ca

Problem 2: Vicious cycles and lock-in effects



The CLD above shows vicious cycles and lock-in effects. The main feedback loops are the following:

- *L1 (+):* **Tasks per person à Work pressure à Resignations à Tasks per person**
  1) This is a very dangerous positive feedback loop – an example of a vicious cycle. When the tasks per person grow up, the work pressure increases too. Depending on the resignation threshold of the people, with the augmentation of the work pressure, it is more likely the members of the team to leave. This increases the tasks per person and makes the situation even worse for a short period of time.
  2) If the managers of the company do not act appropriately, the crisis will go deeper and the whole project may fail.
  3) A solution to this situation is to increase the time until the project's deadline. Then the tasks per person will be reduced and the team members will experience less pressure. It will lead to a situation, in which fewer (if any) resignations will occur. As a result, the company may experience some consequences due to the extended deadline, but the project will not fail.

- *L2 (+):* **Tasks per person à Work pressure à Fatigue à Number of errors – customers' perspective à Quality product – customers' perspective à Customers' satisfaction à Sales à Perceived profit à Team benefits à Motivation à Tasks done à Tasks per person**
  1) This is an example of a positive long-run feedback loop. In a long-run, the work pressure leads to fatigue, which results in the errors that occur in the product. This reduces the quality product (from the customers' perspective). When the product is released, the customers' satisfaction decreases due to the low-quality product. This reduces the sales and the less sales decrease the perceived profit. When the profit decreases, the there are less benefits for the project team, which lead to a lower motivation. As a result, the work is not completed and people have to do deal with even more tasks.
  2) This is a dangerous long-run loop, in case it goes into a wrong direction. The managers of the company will see the result of this loop when the product is released to the market and the first statements of the profit are analyzed.
  3) It is hard to fix such a situation. Depending on some outside factors, some actions can be done:
     i. In order to increase the motivation of the team members, an injection of team benefits might be done. However, this does not remove the reason of the occurred situation. It delays the moment until this situation gets even worse;

ii. The time until deadline can be extended. For example, the team may get some time off in order to recover. This is a temporary solution too since the main reason is not eliminated;

iii. If the customers’ satisfaction is on a low level that influences the sales and the profit remarkably, then the quality control factor should be taken into account. It will reduce the number of errors of the product and the product’s quality will increase. After a certain period of time, the customers’ satisfaction will increase and this will influence the sales and the perceived profit. As a result, there will be more team benefits and members’ motivation, which will help completing the tasks efficiently and will reduce the tasks per person. The team will not feel under pressure and the fatigue will be reduced. Then, the number of errors of the product will be lower, which will increase the product’s quality.

- *<u>L3 (+):</u>* **Available funding à New qualified people à Tasks per person à Work pressure à Fatigue à Number of errors – customers’ perspective à Quality product – customers’ perspective à Customers’ satisfaction à Sales à Perceived profit à Available funding**
  1) This loop shows that the available funding can be invested into new qualified people in order to decrease the tasks per person, the work pressure, the fatigue, and the number of errors. As a result the quality of the product increases. When the high quality product (from clients’ perspective) is released to the market, there will be more satisfied customers. This will increase the sales; the perceived profit will increase too. Then, there will be more available funding that can be invested into new qualified people.
  2) If there is not enough available funding, an outside funding injection may improve the project’s situation a lot in a long period.

- *<u>L4 (+):</u>* **Quality product – customers’ perspective à Customers’ satisfaction à Sales à Perceived profit à Team benefits à Motivation à Tasks done à Quality product – customers’ perspective**
  1) The quality of the product is important. Having a high-quality product (from clients’ perspective) increases the customers’ satisfaction. The perceived profit grows up, which increases the team benefits. Then, the motivation, tasks done, and the quality of the product goes up.
  2) This is another example of a long-run feedback loop which makes customers and team members satisfied and with high morale.

- *<u>L5 (-):</u>* **Tasks per person à Workload per person à Tasks per person**

This is a balancing feedback loop that keeps the work smoothly. However, if the positive loops are dominant, then the influence of this balancing loop will be minimal.

- *L6 (+):* **Quality product – customers' perspective à Customers' satisfaction à Sales à Perceived profit à Available funding à Quality control à Number of errors – customers' perspective à Quality product – customers' perspective**
  1) This is another positive feedback loop, related to the quality of the product. Here, the quality can be achieved by following quality control.
  2) On the other hand this loop can be seen as a lock-in effect:
     If the company does not have quality control, then more errors will occur. This makes the product with low quality. When the product is released to the market, the customers' satisfaction will be low. It will reduce the company's profit and the available funding. With less funding, the company may never start following quality control.
  3) A solution to this vicious cycle could be the injection of outside funding in order to start quality control.

- *L7 (-):* **Risk management à Probability of unexpected events à Tasks per person à Workload per person à Tasks done à Quality product – customers' perspective à Customers' satisfaction à Sales à Perceived profit à Available funding à Risk management**
  This is a balancing feedback loop with regard to risk management. It shows that if there is risk management, then even in the presence of a bad event, the team will continue working smoothly.

- *L8 (+):* **Risk management à Probability of unexpected events à Team confidence à Work pressure à Fatigue à Number of errors – customers' perspective à Customers' satisfaction à Sales à Perceived profit à Available funding à Risk management**
  1) This is another example of a lock-in effect. It shows that if the company does not start doing risk management, it is possible that the company never starts doing it.
  2) The lock-in effect can be avoided by injecting outside funds for risk management. Then the probability of unexpected events will be lower, which will increase the team confidence. The pressure and the fatigue will go down. This will prevent from errors in the product. As a result, a high quality product will be released to the market, which will bring more satisfied customers. The sales will go up; later, the perceived profit will increase as well. Finally, there will be more available funding that can be invested into risk management.

CMPT 816 Problem Set 1

Svetlana Slavova
sds797@mail.usask.ca

Problem 3: Quality attribute tradeoffs

Opportunity cost represents the value of the next best alternative (X) that is not realized, due to the fact that another decision (Y) is made. Opportunity cost is tightly related to trade-off. Trade-off refers to loosing one aspect, when gaining another.

1) Opportunity costs – causing tension in realizing quality attributes within a project

   Opportunity cost may cause tension in realizing different quality attributes. For example, a company is developing a software product. In the ideal case, the product should be time efficient, and at the same time should offer many features (functionality), and to require minimum investment. This leads to the problem that all the characteristics – high speed, full functionality, and low investment level cannot be fulfilled simultaneously. This requires a trade-off between these incompatible aspects of the product.

   If the product's priorities are in the following order:
   1. Speed
   2. Features of the product
   3. Invested money

   Then, the company selects to develop a time-efficient product. As a result only the most the features that are not time-consuming will be implemented. In this case, the opportunity cost of having a time-efficient product is the cost of not implementing all features of the system, plus the consequences of not having those features – such as some clients might not be satisfied due to the reduced functionality of the product.

   If the product's priorities are in the following order:
   1. Features of the product (functionality)
   2. Speed
   3. Invested money

   Then, the opportunity cost of developing a fully-functional system is the cost of not providing a good speed of the product.

2) Technological mechanisms / techniques that could cause trade-offs between quality attributes

1. *Transactional system*
    - Description: A transactional system is able to perform roll-back or roll-forward, i.e. to stay in a stable state, in case of a failure during the system’s transaction. A transactional system may implement flat transactions (ACID transaction properties – atomicity, consistency, isolation, durability) or long-running transactions (SAGAS – consists of a set of flat transactions & compensation actions). An example of a transactional system is a system that consists of a database management system (DBMS).
    - Affected quality attributes: reliability, concurrency control, performance, complexity, locking
    - Why:
        - Since the transactional system is always in a stable state, its reliability is increased;
        - The ACID properties assure consistency of the data, which increases the reliability of the system;
        - The ACID properties assure that there is a high level of concurrency control on the system;
        - The roll-back actions guarantee the recoverability of the system in case of a failure, which increases its reliability;
        - The transactional system allows better performance due to indexing of the data;
        - Many applications and systems are able to share a database, which reduces the redundancy of the data;
        - The transactional technology is more complex than the file technology, due to all features that provides;
        - The transactional system obtains locks over data before processing data and releases it, when the transaction is not committed or rolled-back. This may result in lower performance, if there are many users that need to modify the locked data.

2. *Centralized system*
    - Description: A centralized system consists of a set of non-autonomous components. Usually, it is developed in a homogeneous environment. The centralized system has a point of control. An example of a centralized system is a stand-alone application.
    - Affected quality attributes: reliability, availability, complexity, maintainability

- Why:
    - The single point of control of the centralized system makes it more vulnerable to faults and failures. A failure in one component causes failure in the whole system. This reduces the reliability and the availability of the system;
    - Due to the homogeneous environment and the centralized architecture of the system, the complexity is reduced, as well as the maintainability is achievable relatively easy (compared to decentralized systems).

3. *Distributed system*
    - Description: A distributed system consists of autonomous components (for example Web Services), running on heterogeneous environments, and communicating between each other over a network. Such a system does not have a point of control. A distributed system offers the following properties – scalability, openness, heterogeneity, and fault-tolerance. Client-server applications, peer-to-peer systems, Web Services, as well as the Internet are examples of distributed systems.
    - Affected quality attributes: reliability, scalability, openness, heterogeneity, fault-tolerance, complexity, maintainability, flexibility, loosely-coupled, quality control
    - Why:
        - A distributed system does not have a point of control and respectively of failure. It means that if one component does not work properly at a given time, it may reflect some functionalities of the system, but the whole system will keep running. This increases the system's reliability;
        - The properties scalability, openness, heterogeneity, and fault-tolerance, make the distributed systems more flexible, as well as loosely-coupled;
        - On the other hand, the complexity of the systems increases, as well as their maintenance is more difficult than in the centralized systems;
        - A distributed system requires more quality control, due to its complexity and increased number of points of failures, and at the same time quality control is hard to be done, because of the distributed architecture.

4. *Loosely-coupled system*
    - Description: A loosely-coupled system allows the system's components to be as independent as possible from one another. It means that the

components are autonomous and implement a particular communication protocol. For example, Web Services are loosely-coupled components.

- Affected quality attributes: flexibility, changeability, robustness, performance
- Why:
    - A loosely-coupled system is flexible because one component can be changed with a different one (that provides the same interface and functionality) even at run-time, without modifying the other components of the system, nor stopping the execution of the system.
    - The components of a loosely-coupled system can be distributed across the network easily, which increases the robustness of the system;
    - The loosely-coupled systems require a communication protocol to be used for the communication of the components, which decreases the performance of the systems, compared to tightly-coupled applications.

5. *Duplication technique*
    - Description: Duplication is a technique for improving fault-tolerant systems. A fault-tolerant system is able to continue working properly, even in the presence of one or more failures in its components. It is realized by adding identical components (replicas) to the system that can be executed in parallel or used as a back-up.
    - Affected quality attributes: fault-tolerance, dependability (availability and reliability), system complexity, maintainability, efficiency
    - Why:
        - The availability and the reliability increase, due to the high level of fault-tolerance;
        - The system is more efficient, since the number of failures is kept on a minimum level;
        - The system complexity increases, due to the increased number of components that should be controlled and synchronized;
        - The increased complexity makes the system maintainability more difficult.

6. *RAID technique*
    - Description: RAID is a technique for a data storage that uses multiple hard drives to assure fault-tolerance by replicating data. The RAID disk drives are usually used in servers in order to assure a high level of reliability. The level of lost data is reduced significantly.

- Affected quality attributes: reliability, fault-tolerance, recoverability, data integrity, performance, complexity, cost, transparency
- Why:
  - The nature of the RAID technology assures a high level of dependability, data integrity, as well as reduced disk access time. In addition, the RAID technology provides transparency – from user's perspective, there is just one hard drive, not multiple disks. However, the complexity of disk storages based on this technology increases their cost.

7. *Load-balancing technique*
   - Description: Load balancing is a technique that spreads the work between different components of a system, such as computers and disks. Round-robin is an example of a load-balancing technique.
   - Affected quality attributes: performance, complexity, fault-tolerance, data consistency
   - Why:
     - The performance of a load-balanced system is increased, since there are no components that are overloaded;
     - The technique requires redundancy of components, which makes it more complex, resource-consuming, as well as requires state synchronization, in order to assure data consistency;
     - Load-balancing increases the fault-tolerance of the system because if one component does not work, its load will be distributed among the other components.

8. *CVS (Concurrent Versions System)*
   - Description: CVS is a client-server system that helps keeping track of changes in files of a software project. The goal of the system is to allow programmers to work in parallel on the same project in an easy manner.
   - Affected quality attributes: time efficiency, data consistency, quality control, complexity, speed, applicability
   - Why:
     - CVS reduces time for files synchronization between team members and allows data consistency to be done in an easy manner;
     - CVS assures a level of quality control of the developed product, since the system allows modifications in the source code to be traced, in order to observe what change causes a particular bug;
     - CVS is a complex system that requires network communication, which reflects its speed;

        - ð CVS can be used only for text files, which is a limitation, if the team needs to keep track of changes of other types of files.

9. *Parallelism*
    - § Description: Parallelism is a technique that allows more than one processes to be executed at the same time. This technique can be seen in object-oriented programming languages, as well as in operating systems, realized by threads.
    - § Affected quality attributes: time, resources, quality control, complexity, maintainability
    - § Why:
        - ð The time for tasks execution decreases due to the parallelism;
        - ð The resources can be used more efficiently due to the parallelism;
        - ð The parallelism requires more concurrency control, which increases the necessity of more quality control;
        - ð The parallelism makes the system more complex;
        - ð The system complexity makes the maintainability more difficult.

10. *Black box testing technique*
    - § Description: Black box testing is a technique that is used to check the output of a program, corresponding to a given input. The implementation of the program is not known to the tester. The tester evaluates the program from the end user's perspective in order to observe whether or not a particular program/system conforms to the requirements.
    - § Affected quality attributes: conformance, applicability, accuracy
    - § Why:
        - ð This technique represents testing with respect to the specifications, and as a result, indicates whether or not the requirements of the evaluated product are met;
        - ð The technique can be applied to different levels of the product design – a particular unit, a set of units, the whole system;
        - ð The limited number of inputs that can be tested, reflect the level of accuracy of the technique.