Answer to Question 1

The breadth issues of software complexity are issues associated with the various functionalities of the software. The more such functional units exist, the more breadth issues arise – more complexity and other issues. D2L contains a lot of functional components that handle various functions. Such components result in the breadth issues of software complexity of D2L.

On the other hand, the depth issues of software complexity are the issues associated with the interaction between different functional units of the software. The interaction among those different units is needed to perform various tasks required by the software. These interactions, sometimes, result in issues. Such issues are known as depth issues. D2L functional units are from various domains. Therefore, the interactions between them are always risky: usually result in depth issues.

Answer to Question 2

Problem complexity decomposed for simplicity

D2L allows you to display students registered in a specific course and communicate with them (via email), add a TA to your course, share course materials (links, videos, pdfs, PowerPoint presentations, and other resources needed), gather submitted assignments, mark them, and give feedback online, and create forums for discussion.

Solution software can be decomposed for simplicity

Solutions must be given in step-by-step format by the experts to the students s that they can be comprehendible. In other words, complex and difficult solutions that are not easy to fathom need to be simplified by the experts.

Answer to Question 3

1. The methodologies and processes to be used for development: dividing software development work into distinct phases to improve design, product management, and project management
2. Design and architecture decomposition: high level design and structure of the software system

Answer to Question 4

The number of communication paths for n people is n(n-1)/2.

20*(19)/2 = 190

The complexity of communication increases as the number increases, because there are more channels or pathways through which people can communicate. As the team size increases, communication becomes more complex. To reduce the number of communication paths, we can limit the number of communication paths available. Integration of communication paths can be used to reduce their numbers.

Answer to Question 5

1- The number of expected users
2- The amount of training provided prior to the release
3- The number and type of problems known at the time of release.
4- The expected number of releases that will fix problems and add functionalities

Answer to Question 6

a. Software size increasing to accommodate the rising data in the database.
b. Functional and graphical improvements
c. Without a doubt, the increase in complexity is more difficult to deal with. It makes programming, installing, handline, and using a software more difficult.

Answer to Question 7

Horizontal functions are common to each application feature, such as error processing covering all application features. Horizontal functions and non-functional requirements are associated.

Vertical functions are different application domain-specific functions. Vertical functions and functional requirements are specified.

Components of each function (horizontal and vertical) interact with each other through interfaces.

Answer to Question 8

Integration is bringing smaller components together into a single system that functions as one. It is important, because it can result in an increase in the speed at which information and data are transferred and lower the costs needed.

Answer to Question 9

- Analysis: the software is analyzed in detail to identify the client's requirements.
- Design: The architecture of the project is built/created in this phase. Such architecture includes the standards that need to be set. They are set in this phase.
- Development: during this phase, the software is being developed along with its required components and functionalities.
- Testing: during this phase, the bugs and errors in the software are being assessed.
- Implementation: this phase is vital to ensure the customer's satisfaction. That goal is achieved as the stakeholders assess the developed software.
- Maintenance: during this phase, the software is upgraded every once in a while. That happens when the software passes through the previous stages flawlessly of course

Answer to Question 10

- The process, the people, and the product. The first and last determine the time and number of developers needed for the development. I think the process is the most important, because the product can be replicated by other companies and developers, but the system that the business is built on is how the money is made and how the individuals working have their work (part of the large software) integrated.