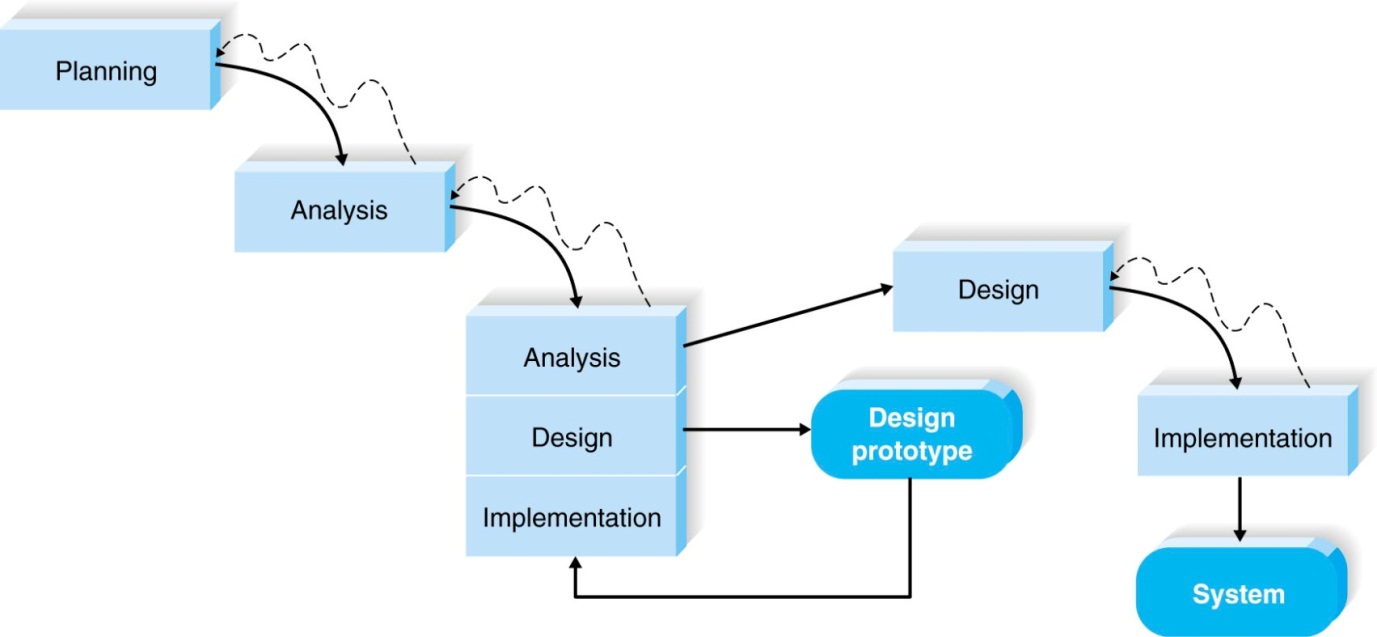
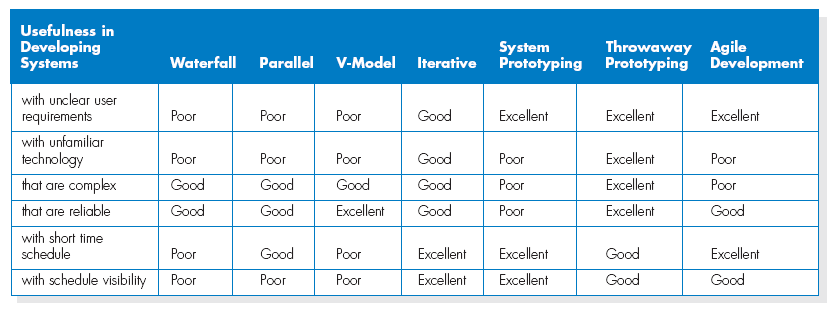
SOFT40091 – Answers to Seminar 2 activities

**1)**



Throwaway prototyping would be a good choice for this scenario for a number of reasons. First, this is a brand new idea, so there may be some ambiguity or confusion as to the functionality of the system. Second, there are technical issues associated with integrating existing hardware and software due to the diversity at different locations around the world. Third, the time frame to delivery is one year.

The time frame would allow for an in-depth analysis to gather information and develop ideas for the system before the design phase. Once the initial requirements were documented, a series of design prototypes can be created, distributed and tested to determine whether issues dealing with functionality or technical problems have been addressed. Once the issues have been resolved, the project can move into design and implementation. See the table below which allows you to compare and contrast the methodologies with the selection criteria discussed in the lecture.



2)

1. How does the Internet, and more specifically the World Wide Web, fit into the picture of systems analysis and systems development?

The Internet has opened up a new frontier for organizations, enabling them to compete on a global basis. This new frontier of electronic commerce is having a tremendous impact on the way that organizations are conducting business. Companies are using the World Wide Web to conduct business with their customers and suppliers, as well as facilitate internal operations. Companies need to reengineer their information systems to take advantage of the World Wide Web and all that it offers. Systems analysis and design is at the heart of this required change.

1. What is Participatory Design?

Participatory Design (PD) is a systems development approach that originated in northern Europe in which users and the improvement in their work lives is the central focus.

1. What are CASE tools? What is a CASE repository and how is it used?

CASE tools provide automated support for some portion of the systems development life cycle. CASE tools include diagramming tools, computer display and report generators, analysis tools, a central repository, documentation generators, and code generators. CASE tools are built around a repository that contains all of the metadata (such as data names, format, uses, and location) for data elements as well as the system specifications. CASE tools automate the repository for easier updating. CASE tools automate routine tasks to help programmers and analysts do their jobs better.

1. Why is it important to use systems analysis and design methodologies when building a system? Why not just build the system in whatever way seems to be “quick and easy”? What value is provided by using an “engineering” approach?

Methodologies, techniques, and tools help ensure the quality and appropriateness of the system being built. Following a systems methodology, applying techniques, and using appropriate tools provides structure to the systems development process, since they have been tested and perfected by others. The quick and easy approach to building systems may be easier, cheaper, and quicker in the short run, but it almost always results in a poorly developed system, meaning that the system will be less than optimal and require extra work to maintain. In the long run, a poorly developed system requires more time and money to make right. Following an engineering-type approach ensures that systems analysis and design is rigorous, structured, and systematic.

1. How is the Joint Application Design (JAD) approach different from the Participatory Design (PD) approach developed? What are the benefits in using approaches like this in building information systems? What are the barriers?

Both Joint Application Design and Participatory Design are development processes designed to help build better systems by engaging the direct participation of users. The primary difference between them lies in the focus of control for systems development. With the JAD approach, control of systems development typically still rests with the systems staff. Indeed, the outputs from JAD sessions are commonly summarized and handled by the systems staff after users have a chance to review the transcripts. With the PD approach, control of systems development is either shared by systems personnel and users or rests solely with the users and their managers. With the movement toward end-user development in the United States, we are moving more toward shared control of systems and systems development. The benefits to the JAD and PD approaches are that they are likely to result in better systems and higher user commitment to the systems than would be the case if these techniques were not used. Some of the barriers to these approaches are that they require extra systems analyst skills and knowledge; in the short run they add more time and expense to the systems development process, and they require more time and effort from already busy users and user managers.