CHAPTER 3: INFORMATION SYSTEM DEVELOPMENT

1&2) What is AGILE system development? How is it different from SDLC?

Agile is an iterative and incremental (evolutionary) approach to software development which is performed:

- in a highly collaborative manner
- By self-organizing teams within an effective governance framework with "just enough" ceremony
- That produces high quality solutions
- In a cost effective and timely manner
- Which meets the changing needs of its stakeholders.

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Agile is based on software development methods where SDLC models like waterfall is based on predictive approach (detailed planning, forecast of tasks and features to be delivered) while agile used iterative approach where there is no planning and clarity on future tasks only in respect to what features needs to be developed, they test frequently in order to minimize major failures.

Customer interaction is the backbone of agile methodology, and open communication with minimum documentation is the typical features of agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

3) What are the reasons to use agile development?

Collaboration

- Agile focus is on people first
- Promotes efficient and face-to-face communication
- Encourages rapid and flexible response to change

Project management

- Maximizes customer value while minimizing waste
- Diminishes technical risk by identifying requirements early
- Contributes to rapidly and cost efficiently adapting to changes

Requirements gathering

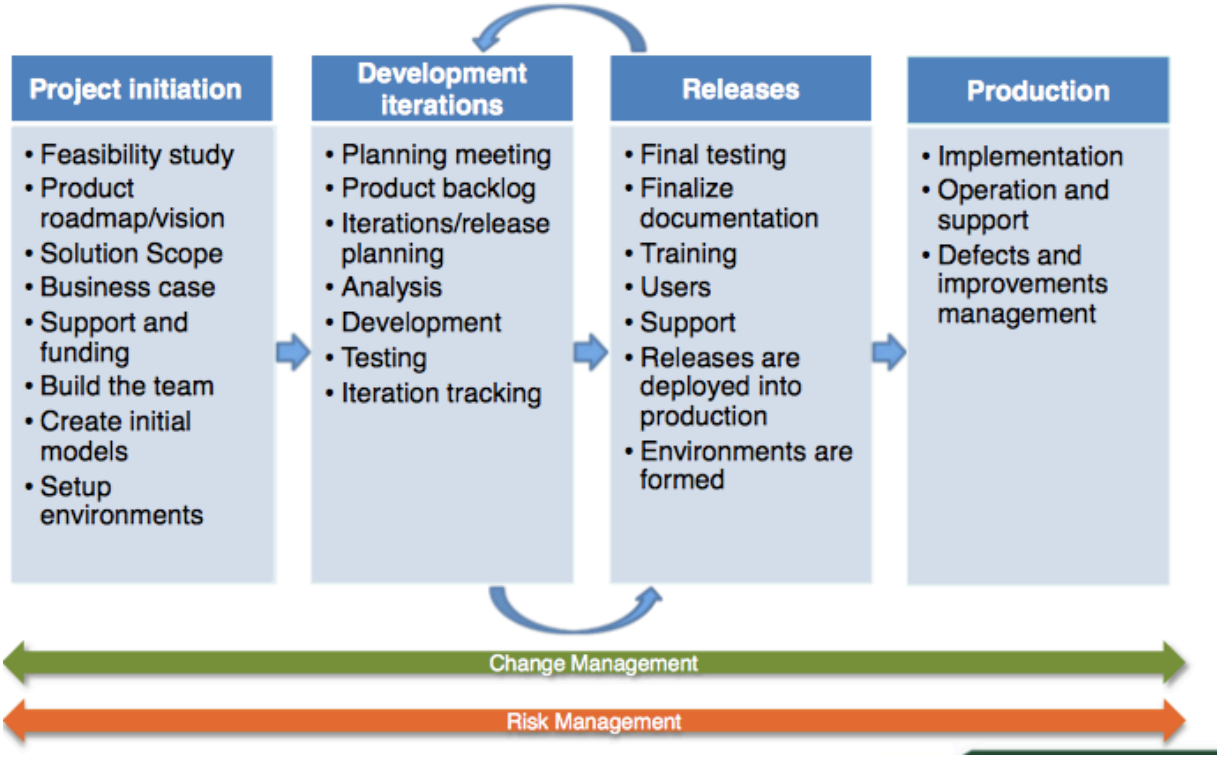
- Ensures progressive elaboration of requirements ensure that end product
- results are consistent with the customer's initial requirements
- Reduces cycle time and increases quality
- Delivers working software frequently which is the primary measurement of success

4) What are the criteria to choose between traditional and agile system development?

- Maturity of the organization?
- Project size and complexity?

- Have similar previous projects been completed successfully?
- Can tested deliverables be completed in short cycles?
- Are requirements for the finished project complete, clear and stable?
- Can the effort required to complete the project be easily predicted?
- What is the level of integration with other systems?
- Time to market deadlines?

5) Briefly explain the phases in agile development lifecycle.



<u>Project Initiation</u>

Your goal during this period is to **initiate** the project by:

1. **Garnering initial support and funding for the project**: You need to be able to provide reasonable answers to questions such as how much is it going to cost, if you're going to get permission to work on the project. You may need to justify your project via a feasibility study

2. **Actively working with stakeholders to** initially model **the scope of the system**

3. **Starting to build the team**: at the beginning of a development project you will need to start identifying key team members and start bringing them onto the team.

4. **Modeling an initial architecture for the system**: Early in the project you need to have at least a general idea of how you're going to build the system (mainframe, COBOL application, .Net application)

5. **Setting up the environment**: you need workstations, development tools, a work area for the team.

6. **Estimating the project**: You'll need to put together an initial **estimate for your agile project** based on the initial requirements, the initial architecture, and the skills of your team. This estimate will evolve throughout the project.

Development Iteration

Agilists incrementally deliver high-quality working software which meets the changing needs of our stakeholders.

Releases
During the release iteration(s), also known as the "end game", we transition the system into **production**.

1. **Final testing of the system**: final system and acceptance testing should be performed at this point, although the majority of testing should be done during construction iterations.

2. **Rework**: there is no value testing the system if you don't plan to act on the defects that you find.

3. **Finalization of any system and user documentation**: documentation is treated like any other requirement: it should be costed, prioritized, and created only if stakeholders are willing to invest in it. Agilists believe that if stakeholders are smart enough to earn the money then they must also be smart enough to spend it appropriately.

4. **Training**: train end users, operations staff, and support staff to work effectively with our system.

5. **Deploy the system**

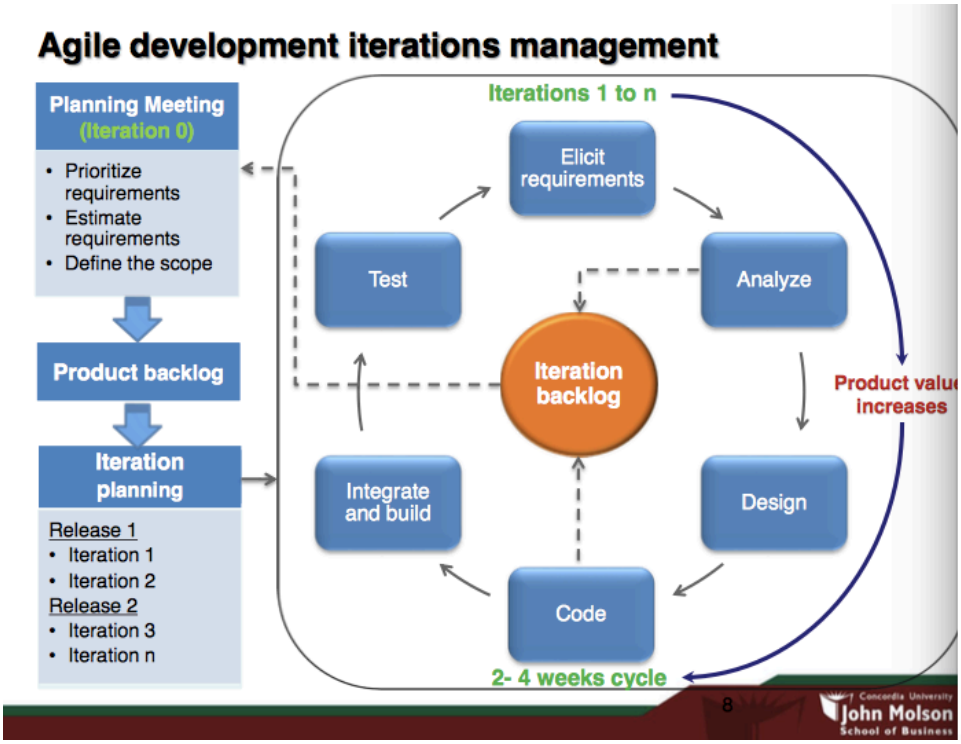
Production

The goal is to keep systems useful and productive after they have been deployed to the user community. This process will differ from organization to organization and perhaps even from system to system, but the fundamental goal remains the same: keep the system running and help users to use it.

6) How is the development iteration managed successfully?

Each iteration implements the highest priority work items, each new work item is prioritized and added to the stack, work items may be reprioritized at any time, work items may be removed at any time.

Coding doesn't begin until design of the entire software application is complete and has gone through a phase gate review. Likewise, testing doesn't begin until coding is complete and has passed necessary phase gate reviews. The purpose of working iteratively is to allow more flexibility for changes.



7) What are the advantages and disadvantages of agile development?

Advantages of Agile development:

- Beneficial in situations where the end-goals of projects are not clearly defined
- Allows for changes to be made after the initial planning (adaptive planning and evolutionary development)
- Sense of ownership from the customer by working extensively and directly with the project team throughout the project
- Customer feedback occurs simultaneously with development in order to respond to changes in requirements rapidly and effectively
- Continuous testing in increments reduces risks related to bugs

Disadvantages of Agile development:

- May be inefficient in large organizations and certain types of projects
- Tend to be hard to estimate time and budget requirements
- Can be quite time consuming
- Possible increase in costs if there is a new for additional iterations
- Team members must be located in the same physical space
- Losing a dedicated resource may be catastrophic

CHAPTER 4: PROJECT MANAGEMENT

1) What is systems view of project management and why is it important?

To view a project with a systems perspective is to see the whole internal and external operational landscape in which the project will be functioning.

It is critical to successful project management (process of scoping, planning, staffing, directing and controlling the development of an acceptable system at a minimum cost within a specific time frame)

Systems management involves managing the business, holistic, technological and organizational issues associated with making a change to the system

- Need to identify key business, technological and organizational issues
- Ensure key stakeholders satisfaction
- Ultimately do what is best for the organization

2) What are the conditions of a project to be considered a success?

A project is considered a success if

- Accepted by the customer
- Delivered on time
- Delivered with budget
- System development process had minimal impact on business operations
- Meets the desired level of quality

3) What are the functions of project management?

Scoping the project: defining the boundaries of the project (assumptions, costs, expectations)

Planning project tasks and staffing the project team: Tasks to complete the project. Combination of scope and methodology

Estimating required resources: People and duration vs skills, tasks sequencing

Scheduling activities: Tasks, tasks durations and task prerequisites

Organizing the project effort - (RACI): Roles and responsibilities of team members

Directing the project team's activities**:** Team activities. People management (coordinate, motivate, delegate, reward)

Controlling the project: The most difficult function

Closing the project: Assessing successes and failures (lessons learned)

4) What skills do project managers need?

1. Personal-Soft skills (motivate and sustain people, address and solve problems within the team, as well as outside the team)

– Positive attitude
– Be considerate
– Conflict management

Why is this important?
– To motivate and sustain people
– In order to address problems within the team and outside as well
2. Technical skills
– Appropriate level based on type, size, structure of the project, etc.
– Project manager is responsible for the management of the project

3. Management skills
– Organization, communication, finance and HR

4. Coping skills involves:
   - Flexibility
   - Persistent but firm
   - Receive stress and deal with it (protect the team)

5. Political and cultural awareness

5) What are the activities in project management lifecycle?

1. Negotiate scope: all parties must agree to the project scope before any attempt is made to identify and schedule tasks or assign resources (people) to those tasks

2. Identify tasks:

3. Estimate tasks duration

4. Specify tasks dependencies

5. Assign resources

6. Direct team effort

7. Monitor and control progress

8. Assess project result and experiences

6) What responsibilities project managers have?

1. Managing problems (five step problem solving model)
   - Understand the problem
   - Define the root causes
   - Determine solutions
   - Decide and plan
   - Implement and evaluate

2. Managing risks
– Assess project risk (What, How, If it happens and consequences)

– Manage project risks (Mitigation mechanisms to avoid consequences)

3. Managing conflicts
– Identify the sources of conflict among stakeholder project stakeholders – Identify conflict management resolutions

4. Sustaining commitment to projects
– Determinants (psychological, social and organizational)

5. Managing communications among project stakeholders
– Common project vocabulary and assigning a facilitator

6. Managing time
7. Managing costs
8. Managing quality

7) What are the project mismanagement problems and their consequences?

- Lack of organization's commitment to the system development methodology
- Taking shortcuts through or around the system development methodology
- Poor expectations management (scope and feature creep)
- Premature commitment to a fixed budget and schedule
- Poor estimating techniques
- Over optimism
- The mythical person-month
- Inadequate people management skills
- Failure to adapt to business change
- Insufficient resources
- Failure to "manage to the plan"
- Major cause: Project manager education and training

8) What are project manager competencies?

<u>Business achievement</u>
– Ties project to mission, vision and goals
– Stakeholders involved
– Quality expectations

<u>People management</u>
– Motivation
– Communication (writing, oral, presentations and meetings)
– Develop others (training, coaching, feedback, mentoring, performance reviews)

<u>Self management</u>
– Works effectively under pressure
– Integrity, honesty and respect
– Flexibility in order to adjust to new situations (unforeseen/unanticipated)

<u>Influence</u>
– Interpersonal relationships (different profiles)
– Risk and expectations management

– Obtain consensus of stakeholders

9) What are project management tools and techniques

Techniques

PERT (Project Evaluation and Review Technique)
– Graphical network used to display the interdependencies between project tasks
– Scheduled start-finish and actual start-finish

Gantt chart
– High level project scheduling and progress
– Horizontal bar that depicts project task against a calendar – Simple to understand
– Visibility on overlapping tasks

Software tools most commonly used

- Microsoft project
- Primavera

CHAPTER 5: SYSTEM ANALYSIS

1) What is the Business Analyst approach according to BABOK guide?

Business analysis approaches describe the overall method that will be followed when performing business analysis work on a given initiative, how and when tasks will be performed, and the deliverables that will be produced.

The business analysis approach should:

- align to the overall goals of the change,
- coordinate the business analysis tasks with the activities and deliverables of the overall change,
- include tasks to manage any risks that could reduce the quality of business analysis deliverables or impede task efficiency, and
- leverage approaches and select techniques and tools that have historically worked well.

2) What are the different approaches to system analysis?

1. Model-Driven Analysis Approaches (uses pictures to communicate business problems, requirements and solutions):

- Structured analysis (uses process modeling – DFD's): focuses on the flow of data through business and software processes. Shows the flow of data between and through processes and shows the place where data is stored.

- Object-oriented analysis (uses object modeling - use cases): views information systems not as data and processes but as a collection of objects that summarize data and processes. (Java, C++)

2. Accelerated Analysis Approaches (emphasize the construction of prototypes to more rapidly identify business and user requirements for a new system):

- Discovery Prototyping (Access): technique used to identify the user's business requirements by having them react to a quick implementation of those requirements. Involves building small-scale subsystems to discover requirements.

- Rapid Architected Analysis (RAD, requires reverse engineering): accelerated analysis approach that builds system models. Automatically generates a system model from prototype= read code to generate.

3) Differentiate between logical and physical system models?

Logical: deals with gathering business requirements, and converting those requirements into a model.

Physical: conversion of the logical business model into a relational database modeling.

4) Describe each phase within systems analysis in terms of objectives, tasks, and stakeholders? Re read part in book

1. Scope definition:
   Objective: determines worthiness of the project.

Tasks: identify baseline problems and opportunities, negotiate scope, assess project worthiness, develop schedule and budget and communicate the project plan.
Stakeholders: system owner, committee, all stakeholders if necessary.

2. Problem analysis phase:
   Objective: answers questions.
   Tasks: understand the problem domain, analyze business processes, opportunities and problems, establish system improvement objectives, update the project plan, and communicate findings and recommendations.
   Stakeholders: project manager, system owner, system user.

3. Requirement analysis:
   Objective: identify what the new system is without considering technology.
   Tasks: Identify and express system requirements (functional and non function=performance, prioritize system requirements-time boxing, update or refine the project plan and communicate the requirements statement.
   Stakeholders: system analyst with users and owners.

4. Logical design phase: Stakeholders = system analysts and appropriate participants.
   Objective: further documents business requirements using system models that illustrates data structures, business processes, data flows and user interfaces.
   Tasks: structure functional requirements, prototype functional requirements, validate functional requirements and define acceptance test cases.
   Stakeholders:?

5. Decision making phase:
   Objective: transition the project from business concerns to technical system solutions.
   Tasks: Identify candidate solutions (candidate systems matrix), analyze candidate solutions (feasibility matrix), compare candidate solutions, update project plan and recommend a system solution.
   Stakeholders: ?

5) What is analysis paralysis? How can it be prevented?

In the problem analysis phase: task understands the problem domain = beware of analysis paralysis = overanalyzing a system until the decision is never taken or with delay then paralyzing the outcome.

6) Are user's requirements frozen for the duration of system development as they are stated in the requirement statements?

User's requirements are not frozen for the duration of system development, we live in a fast pace economy and we need to adapt to changes. Requirement management defines a process for system owners, users, analysts, designers, and builders to submit proposed changes to requirements for a system. The process specifies how changes are to be requested and documented, how they will be logged and tracked, when and how they will be assessed for priority and how they will eventually be satisfied.

CHAPTER 6: REQUIREMENT DISCOVERY

1) What is discovery prototyping (fact-finding technique)? Advantages/Disadvantages?

- The act of building a small scale-working model of the user's requirement in order to discover or verify those requirements.
- Discovery prototyping is frequently applied to systems development projects, especially in cases where the development team is having problems defining the system requirements.

Advantages:

- Allow users and developers to experiment with the software and develop an understanding of how the system might work.
- Aids in determining the feasibility and usefulness of the system before high development costs are incurred.
- Serves as a training mechanism for users.

Disadvantages:

- Developers may need to be trained in the prototyping approach.
- Users may develop unrealistic expectations based on the performance reliability and features of the prototype.
- Doing prototyping may extend the development schedule and increase the costs.

2) What is Joint Requirement Planning (JRP)? Who are the participants?

- A process whereby meetings are conducted for the purpose of analyzing problems and defining requirements.
- Strategy in which all stakeholders attend an intensive workshop aimed at reaching consensus on project decisions meaning scope, budget, schedule. (1-3 days)

Participants:

- Sponsor: top management who has authority that spans the different departments and users who are to be involved in the systems project.
- Facilitator: responsible for leading all sessions that are held for a systems project.
- Scribe: responsible for keeping records pertaining to everything discussed in the meeting.
- Users and managers: a number of participants from the user and management sectors of an organization who are given release time from their day-to-day jobs to devote themselves to active involvement in the JRP sessions (chosen by the sponsor).
- IT staff: members of the project team that work closely with the scribe to develop models and other documentation related to facts communicated during the meeting.

3) Benefits and limitations of JRPxs

Benefits:

- Encourages user and management ownership of the project
- Reduces system development time
- May enjoy the benefits of prototyping

Limitation:

- Success depends on the leader's abilities.

4) How are JRP sessions planned and conducted?

Planned:

1. Selecting a location for JRP sessions: away from the company workplace in a conference room
2. Selecting JRP participations (listed above): users should be key individuals who are knowledgeable about their business area.
3. Preparing a JRP session agenda: facilitator must prepare documentation to brief the participants about the scope and objectives of the session.

Conducted:

- Do not unreasonable deviate from the agenda
- Stay on schedule
- Scribe takes notes
- Avoid jargon
- Encourage participation

Goals: is to generate possible ideas to solve a problem

CHAPTER 11: FEASIBILITY ANALYSIS AND THE SYSTEM PROPOSAL

1) What is "Solution Evaluation" according to the BABOK Guide?

Solution Evaluation describes tasks that analyze the actual value being delivered, identifies limitations which may be preventing value from being realized, and makes recommendations to increase the value of the solution. It may include any combination of performance assessments, tests, and experiments, and may combine both objective and subjective assessments of value. Solution Evaluation generally focuses on a component of an enterprise rather than the entire enterprise.

2) What are the feasibility checkpoints in the systems analysis phase of SLDC?

Systems analysis phase:

- Scope definition checkpoint: conducted during the scope definition phase. Feasibility can't be measured before problems/opportunities and requirements are fully understood.

- Problem analysis checkpoint: occurs after a more detailed study and problem analysis of the current system. Development costs are still just estimates and analysists have yet to fully define user requirements or to specify a design solution to those requirements.

System design phase (estimate feasibility):

- Decision making checkpoint: problems and requirements should be known by now.

3) What is creeping commitment approach of feasibility?

The scope and complexity of an apparently feasible project can change after the initial problems and opportunities are fully analyzed or after the system has been designed (at any of the checkpoints the project may be cancelled, revised or continued).

4) What are the tests for feasibility?

- Operational feasibility: a measure of how well a solution meets the identified system requirements to solve the problems and take advantage of the opportunities envisioned for the system.

- Cultural (or political) feasibility: a measure of how well the solution will be accepted in a given organizational climate.

- Technical feasibility: a measure of the practicality of a technical solution and the availability of technical resources and expertise.

- Schedule feasibility: a measure of how reasonable a project timetable is.

- Economic feasibility: a measure of the cost-effectiveness of a project or solution.

- Legal feasibility: a measure of how well a solution can be implemented within existing legal and contractual obligations.

5) What categories of cost are considered in assessing the cost of a system?

Determining costs (categories):

- Personnel costs: the salaries of system analysists, programmers, consultants etc.
- Computer usage costs: programming, testing, conversion etc.
- Training costs: if computer personal or end users have to be trained.
- Supply, duplication and technical costs
- Cost of any computer equipment and software

6) What are the type of costs and benefits considered in cost benefits analysis?

Determining costs:

- Fixed costs (development): occur at regular intervals but at relatively fixed rates (lease payments and software license payments)

- Variable costs (operation): occur in proportion to some usage factor (costs of computer usage, supplies, utilities and maintenance)

Determining benefits:

- Tangible benefits: those that can be easily quantified (usually measured in terms of monthly or annual savings or of profit to the firm)

- Intangible benefits: those that are believed to be difficult or impossible to quantify (improved customer goodwill, improved employee morale and better service to community)

7) Candidate system matrix

It is a useful tool for documenting the similarities and differences between candidate systems being considered (offers no analysis).

The structure of the matrix:

FIGURE 11-6 Candidate Systems Matrix Template

	Candidate 1 Name	Candidate 2 Name	Candidate 3 Name
Stakeholders			
Knowledge			
Processes			
Communications			

The columns represent candidate solutions.
Rows of the matrix represent characteristics that differentiate the candidates.

Stakeholders: identify how the system will interact with people and other systems.
Knowledge: identify how data stores will be implemented.
Processes: identify how manual business processes will be modified.
Communications: how processes and data will be distributed.

Approaches to identifying candidate solutions:

- Recognizing user's ideas and opinions: throughout a systems project, users may suggest manual or technology-related solutions (they should be given consideration).

- Consulting methodology and architecture standards: dictate how technology solutions are to be selected and what technologies may be represented.

- Brainstorming possible solutions: effective technique for identifying possible solutions

7) Feasibility analysis matrix

A tool used to rank candidate systems

The structure of the matrix:

FIGURE 11-9 Sample Feasibility Analysis Matrix

	Wt	Candidate 1	Candidate 2	Candidate 3
Description		Purchase commercial off-the-shelf package for member services.	Write new application in-house using new company standard VB. NET and SQL Server database	Rewrite current in-house application using Powerbuilder.
Operational feasibility	15%	Supports only Member Services requirements. Current business process would have to be modified to take advantage of software functionality. Also, there is concern about security in the system. **Score: 60**	Fully supports user-required functionality. **Score: 100**	Fully supports user-required functionality. **Score: 100**
Cultural feasibility	15%	Possible user resistance to nonstandard user interface of proposed purchased package. **Score: 70**	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**
Technical feasibility	20%	Current production release of Platinum Plus package is version 1.0 and has been on the market for only 6 weeks. Maturity of product is a risk, and company charges and additional monthly fee for technical support. Required to hire or train Java J2EE expertise to perform modifications for integration requirements. **Score: 50**	Solution requires writing application in VB. NET. Although current technical staff has only Powerbuilder experience, it should be relatively easy to find programmers with VB. NET experience. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned about acquisition of Powerbuilder by Sybase Inc. MS SQL Server is the current company standard for database, which competes with Sybase DBMS. We have no guarantee that future versions of Powerbuilder will "play well" with our current version of SQL Server. **Score: 60**
Economic feasibility	30%			
Cost to develop:		Approx. $350,000	Approx. $418,000	Approx. $400,000
Payback (discounted):		Approx. 4.5 years	Approx. 3.5 years	Approx. 3.3 years
Net present value:		Approx. $210,000	Approx. $307,000	Approx. $325,000
Detailed calculations:		See Attachment A **Score: 60**	See Attachment A **Score: 85**	See Attachment A **Score: 90**
Schedule feasibility	10%	Less than 3 months **Score: 95**	9–12 months **Score: 80**	9 months **Score: 85**
Legal feasibility	10%	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**
Weighted score	100%	67	92.5	87.5

Who provides the data for the matrix?
What happens if there is difference of opinion about the input to the matrix?