1. How does SWEBOK address the software component of software project management?

SWEBOK stands for Software Engineering Body of Knowledge, as the name suggests, this is the standard's body for Software Engineering as a whole. Software according to SWEBOK is the application of systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. Simple put, it is the application of engineering to software. So how do we assess our software engineering knowledge and skills? In part, that is what the guide to the SWEBOK is about. It identifies the knowledge areas and provides guidance to book references that covers those knowledge areas for what is accepted as general practice for software engineering.

SWEBOK identifies the discipline of engineering a software into ten different knowledge areas (KA's);

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods
- Software quality

2. How does PMBOK address the *project management* component of software project management?

Project Management Body of Knowledge (PMBOK) is an inclusive term that describes the sum of knowledge within the profession of project management (including software project management). Project Management according to PMBOK is the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project. This involves balancing competing demands among scope, time, cost, quality, needs and expectations of stakeholders. In regard to software project management, PMBOK provides an extensive support relevant to the software engineering management with a notion of project management, as the construction of useful software artifacts which is normally managed in the form of individual projects. PMBOK includes several project management knowledge areas (KAs) that have direct relevance to software project management, these KAs include project integration management, project scope management, project time management, project cost management, project quality management, project human resource management, project risk management and project communications management.

Software project management is an essential part of software engineering. In general a software project management includes management activities, project planning, project scheduling and project risk management. Each of these tasks and practices are tailored to meet the needs of a specific project. The nine knowledge areas identified in PMBOK addresses the project management component of software project management by describing the project management knowledge and practice in terms of its component processes. Although much of the knowledge needed to manage a software project is unique or nearly unique to software project management, PMBOK indentifies a general management that encompasses planning, organizing, staffing, executing, and controlling the operations of an ongoing enterprise.

3. How can you utilize the following tools and techniques for analysis, modeling, and management throughout your project development phases:

- **Acclaro DFSS**

This tools aims to fill the gap between customer requirements and what needs to built to satisfy those requirements. This is to be used by Business Analysts in the earlier stages of a project to gather what the customer wants (Functional Requirements: FR's). These FR's are later used to drive the design of the application/solution.

It has the ability to show the design parameters (DP's) and FR's in a matrix like layout making it easy to understand the relationship and interdependency of the two. This is useful to identify quickly which of the customer requirements are high priority and which take a long time to implement. The information can then be used by the stakeholder's early on in the project to determine which of the requirements should be complete to meet the budget.

- **Microsoft Project**

Microsoft Project is very well known in the industry of project management. It is practically a standard in its field and one that is used to measure other up coming tools. Although tools has many capabilities and more get added with each release, it is not geared toward analysis and modeling of a project specifically during the development phases. This product is very attuned to helping a Project manager in establishing a plan for the project at hand, allocating resources to tasks in the plan, monitoring the progress of the task as well as the plan as a whole and forecasting the stages of the project on a timeline.

Our particular interaction with MS Project has been mainly to identify and organize the tasks in the our project plan. Assign resources to the tasks as well as estimated duration associated with individual tasks. This quickly organized the tasks into categories and enabled us to view the complete timeline. Project also does a surprisingly good job at determining the budget for the project when the resource cost information was provided.

This is a invaluable tool for any project manager.

- **C-map tools**

This is one of several applications, in what appears to be a very hot market of tools to replace the simple 'paper and pencil' in a brainstorming session. Let face it we have all in the project management or development field have brainstormed with colleagues using 'paper and pencil' or whiteboard which every is accessible. Some of the brainstorming session go undocumented and others are completely lost leading to mistakes or rework.

C-map fits into this category very comfortably with 'ease of use' as its biggest strengths, followed closely by 'zero cost' and 'export to HTML' features. We initially used c-map to document our brainstorming session for our project 'ShareIt.com'. The main idea behind the site and what the features are going to be offered as part of it. Now the c-map has morphed into the main repository for all our documentation and collaboration on the project. We have also documented our website page flow in c-map for quick reference. As mentioned earlier, c-map and similar tools cannot be categorized into a certain area of software project management, they can be used for anything and everything depending on what is needed.

- **Visio/Telelogic for OO modeling using UML/SysML**

Visio, another Microsoft application, is well suited for visualizing infrastructure or connection/interactions between different nodes of a software based application. It provides a way for the project manager to quick document for the rest of the group, how the project environment is going to be laid out. Another feature to note is DB visualization, this particular feature enabled Visio to connect to any database via ODBC connection and retrieve the schema structure along with the relationships between the tables.

Although the product is advertised as such, it does not support UML very well. It can be used for UML design, but that would need more ingrained knowledge of the tool and UML concepts. Definitely not the standard for UML modeling, often used within projects to document the infrastructure decision and changes.

- **Weekly project log and periodic status reports**

A project log or periodic status updates both exist in any given project to serve one purpose. To update the manager and other project stakeholders of the progress of individual tasks, any issues that have come up which might potential effect the timeline and/or budget of the entire project. Project log are less common than status update meetings. You often notice that the project logs are maintained/updated after the fact so that others outside the immediate project group can keep informed.

In my experience, the type of project status updates that are most effective that once specified by the RUP (Rational Unified Process) or more accurately AUP (Agile Unified Process). In either of these processes the project manager is supposed to have a short standing project update meeting every morning for the duration of the project to check on status of the items being worked on. In these meeting everyone involved is given a chance to report on the status of their task and indicate 'on schedule', 'behind' or 'ahead'. This, in my view accomplishes a couple of very important things; 1. Removes the burden of maintaining a project log or report by the resources that are accomplishing the task. 2. Any critical issues are arise are brought to the attention of the project manager immediately and any resource allocation issues can be mitigated quickly.

4. Before Spring Break, the emphasis of the course was on SWEBOK and defining your project by assigning group roles and planning your approach using various architecting and modeling techniques for risk mitigation. Why is it important for software project managers to understand some aspects of SWEBOK such as information architecture development in order to manage effectively in the software domain?

à Information architecture development refers to the art of expressing a model or concept of information used in activities that require explicit details of complex systems. The software architect is responsible for articulating the architectural vision, conceptualizing and experimenting with alternative architectural approaches, creating models and components, interface specification documents, and validating the architecture against requirements and assumptions. It becomes important for software project managers to understand some aspects of SWEBOK such as information architecture development to effectively manage in software domain because the information and system architecture affects the performance, robustness, distributability and maintainability of a system.

Before Spring Break, considering the progress of our project,’ShareIT.com’, C-map and other tools were used to define the web design architecture that showed how the Web project is decomposed and organized into components – and the interfaces between those components. The information architecture for our project is expected to include several deliverable including Hypermap (A conceptual representation of the user flow through the site), Site Map (A graphical representation of the site structure and navigation, showing how many screens there will be and how they link to one another), System Architecture Components Model (Provides a summary of the major technical components of the system and how they will interact at a high level), Component-Based Requirements Descriptions (A functional view of the system requirements that maps directly to the object model and the hypermap, thus integrating technical, business, and user experience requirements), etc. These deliverables plays important role in developing the software (Web Site for our project), allows the team to analyze the models and the solutions to be implemented, analyze if the requirements are going to meet the shareholders needs and expectations and plan the subsequent developing activities such as testing and maintenance, etc. Hence for a successful completion of a project with given scope, budget, schedule and quality it is important for the project manager to understand and integrate some aspects of SWEBOK, depending of the needs of the project.

5. After Spring Break, the emphasis of the course is on PMBOK and learning how to manage various aspects of your project architected and modeled before Spring Break. Please select 1 or 2 key chapters from PMBOK which you have reviewed on C-map and describe any insight you may have gained that can help you improve your project management approach in your project and/or industry.

According to the Project Management Body of Knowledge (PMBOK), project management is the application of knowledge, skills, tools and techniques to project activities to meet project requirements. Project management is accomplished through the application and integration of the project management processes of initiating, planning, executing, monitoring and controlling, and closing. The tasks involved in managing a project include, but are not limited to: identifying requirements; establishing clear and achievable objectives; balancing the competing demands for quality, scope, time, and cost; adapting the specifications, plans, and approach to the different concerns and expectations of various stakeholders.

The two chapters reviewed from the PMBOK are chapters 5 "Project Scope Management" and 6 "Project Time Management". These two chapters were chosen because they represent the areas that engineers must deal with the most. While the other areas of project management may be equally important, software engineers must handle scope and time management issues on a daily basis.

Project Scope management included the processes required to ensure that the project includes all the work required, and only the work required to complete the project successfully. It involves the following knowledge areas: Scope Planning, Scope Definition, Create WBS, Scope Verification, Scope Control.

Project Time Management included the process required to accomplish the timely completion of the project. The processes include the following: activity definition, activity sequencing, activity resource estimating, activity duration estimating, schedule development and schedule control.

The areas covered by these two chapters are very helpful in controlling the project. First, Scope Management, gives ideas on how to keep the scope of the project under control by using the techniques outlined in the PMBOK. The Time Management area has proved useful for defining the activities and sequencing them in a logical manner.