Textbook Assignment 4

1. How many process areas, in total, are included in SEI's Software CMMI? List those that fall into the engineering category and the support category.

There are, in total, 25 process areas in CMMI.

Under the engineering category:

- Requirements development
- Requirements management
- Technical solutions
- Product integration
- Verification
- Validation

Under the support category:

- Configuration management
- Process and product quality assurance
- Measurement and analysis
- Organizational environment for integration
- Decision analysis and resolution
- Casual analysis and resolution

2. List the process areas that are required for staged maturity level 2 of CMMI. How do these differ from those of maturity level 2 in CMM?

Process areas required for staged maturity level 2:

- Requirements management
- Project planning
- Project monitoring and control
- Supplier agreement management
- Measurement and analysis
- Process and product quality assurance
- Configuration management

An organization at Level 2 (the repeatable level) must master the following key project management-related processes and is expected to be able to repeat its success when given a similar project:

- Requirements management
- Software project tracking and oversight
- Software quality assurance
- Software project planning
- Subcontract management
- Software configuration management

3. Discuss the two representation models in CMMI. What do these two models assess? In the continuous representation model, discuss how a process area moves up (or improve) from CL2 to CL3.

- Staged representation model: there are 5 maturity levels. Focuses more on process improvement because all process areas will have a specific goal. Assesses the maturity of an organization.
- Continuous representation model: more applicable to the assessment and improvement of processes. Instead of maturity levels, continuous representation model uses 6 capability levels for assessing the process areas.

4. List the four core values of XP, and five of the 12 XP practices.

Four core values of XP:

- Frequent communication between the team members and with the customer
- Simplicity in design and code
- Feedback at many different levels. Unit tests and continuous integration provide feedback to the individual developer, or pair of developers. Also, small iterations provide customer feedback
- Courage to implement hard but necessary decisions. One possible decision is to not use XP, if it does not seem appropriate for the project.

Five of the 12 XP practices:

- Planning
- Short releases
- Metaphor
- Simple design
- Test-driven development

5. Explain the characteristics of Agile methodologies.

Using Agile methodologies, when we are talking about the requirements, we assume they will change; requirements collected informally at the beginning of the project, and then at the beginning of each iteration. It implements constant user interaction instead of formal requirements.

When talking about planning, we do not plan too much up front. Planning is done in small increments throughout the development.

When talking about scheduling, only the next few activities are scheduled. Schedule may change if scope needs to be adjusted

When considering the design, it is informal and iterative.

For the user-involvement, it is crucial, frequent, throughout the whole process.

Documentation is minimal, only what is necessary; relies on source code as the ultimate documentation.

Communication is done informally, throughout the project.

The process complexity is relatively low; initial description contains less than 200 pages.

Low overhead.

6. List three advantages of Agile over traditional models and explain why. List three advantages of traditional models over Agile and explain why.

Advantages of Agile:

- Low process complexity: the processes themselves are simple, which allow them to be easily understood and implemented.
- Low cost and overhead: the processes mandate very few activities that do not directly produce software.
- Efficient handling of changes: the processes are designed with the assumption that requirements will change, and the methodology is prepared for these changes.

Advantages of traditional models:

- They are scalable: they are better suited to larger projects. They could be scaled up the largest projects and can be scaled down for smaller projects. Unlike Agile processes that mostly are not able to be scaled without losing some or all their agility
- Little reliance on teamwork: roles will be defined. This can be appropriate for most kinds of people because they do not require tight team playing; almost any personality will work as long as the team members can follow the rules set.
- Cultural clashes: They are better suited for larger companies with possibly geographically remote sites and more formal cultures

7. What is test-driven programming, and which Agile process advocates it?

Test-driven programming, also known as test-driven development, is a method of creating software programs in which we ensure that testing is done continuously and is automated as much as possible. We write unit tests for all code. In certain situations, test-first development is performed. We write the tests before we write the actual code. Keep running tests all the time. Ask customers to write functional acceptance tests for verify when the features are finished. Continue to keep these tests running after they run the first time. The Agile process that advocates test-driven programming is XP or Extreme Programming.

8. When we "pull" in a software development process, what are we pulling?

We are pulling the needed components from upstream processes in a just-in-time fashion using a card to specify what parts are needed to complete the downstream task.

9. Explain why Scrum is not a "pure" agile model.

- Agile processes are a family of software development methodologies that produce software in short iterations and allow for greater changes in design. Usually work as a team, one unit.
- Scrum is an Agile methodology that has demonstrated good results. Scrum framework consists of team roles, events, artifacts, and rules. There are 3 core roles that are vital to the success of the project: the product owner, the development team, and the Scrum Master.

10. Consider a large, industry-specific product such as a hospital management system for a group of hospitals, Discuss the pros and cons of using Agile methodologies versus a traditional process where more rigid planning and documentation are required. Focus on issues such as project size, team size, continuing multiple releases over years, expanding the project for an international market, and worldwide customer support.

Pros of Agile:

- Low process complexity: the processes themselves are simple, which allows them to be easily understood and implemented
- Low cost and overhead: the processes mandate very few activities that do not directly produce software.
- Efficient handling of changed: the processes are designed with the assumption that requirements will change and the methodology is prepared for such changes.
- Fast results: most agile processes have fast iterations and produce a core system that can be used in a relatively short time. The system will then be improved and more functionality added as the project progresses. Given that the processes have low overhead, they also tend to produce final results after it is made of continuous integration

- Usable systems: because the customer is involved and the process deals well with changes, chances are the final product will be what the customers actually want when the project is completed rather than what was originally planned as requirements.

Cons of Agile:

- Possibly not scalable: Agile processes have been utilized by relatively small teams, and are not able to scale without losing some or a lot of their agility. Many projects are too big or too critical to be developed with Agile methods. Regardless, many of the Agile practices, ideas, and principles can be incorporated into traditional methodologies.
- Heavy reliance on teamwork: not all people are able to work well in teams. Often, one bad member can destroy the cohesiveness of the entire team. Agile methods rely on informal communication and team dynamics much more than traditional ones.
- Reliance on frequent access to customer: Agile methods need frequent feedback from the customer, the on-site customer in XP, and the small releases with feedback from customers. This level of constant customer feedback will simply not work when a team is set to develop a large enterprise application such as PeopleSoft or SAP. In these large industry-wide applications, multiple industry experts across a span of ten to twenty companies are involved in the requirements process. Because these interactions with the customers are not free, they have to be well planned and coordinated ahead of time. Not all customers are willing or able to provide this level of cooperation and feedback. Without customer feedback, Agile methods are not able to validate requirements or adapt to change. Traditional methodologies concentrate the feedback at the beginning of the process during the requirements phase and at the end of the process during the acceptance tests phase
- Cultural clash: many XP practices clash with accepted software engineering wisdom, or with common management techniques. Performance evaluations, for example, are harder to perform in XP because the work is done in pairs, and the code is owned by the whole team.

The size/scope of Agile(XP) is small; limited to one team of up to ten people. While the scope of traditional (RUP) is better suited for larger projects. The criticality of Agile(XP) is relatively low; not suitable for life-critical systems without adaptation. That of traditional (RUP) can be used for mission critical systems (maybe with minimal modifications). When it comes to people, Agile(XP) is more suitable for team players, “good citizens” who can do design and programming adequately. XP requires strict adherence to certain practices. On the other hand, in Traditional (RUP), many roles are defined which can be appropriate for most kinds of people. Therefore it doesn’t require team playing. Almost any personality will work as long as the team member can follow rules. The company culture for Agile(XP) is better suited for small co0located companies with relaxed cultures. That of traditional (RUP) is better suited for larger companies with possibly geographically remote sites and more formal cultures. The stability of Agile (XP) copes easily with changes in requirements or environment. While that of Traditional (RUP) is less suited to cope with changes. Assumes a relatively stable environment where requirements don’t change much. Can be adapted.

Now when considering the above case (in the question) the environment will be bigger than the usual environment where we would use Agile methodology effectively. Therefore, it would only make sense to use the Traditional Methodology.