A brief description of the problems faced by a hypothetical software company is presented below, as narrated by the Process Manager. This narration defines the situation for which you are required to develop a bespoke software development methodology.

**Process Manager:** Our problems started from the very first day, and promise to continue for the foreseeable future. The business endeavors of our company, Complex Dynamics, are not fully successful, but they are not total failures either. We are taking on an ever-increasing number of projects, and the rate of failed or unfinished projects is rising alarmingly. Several projects are underway at any one time, on each of which several teams are working simultaneously (each team is made up of up to 10 people). There is a tendency on the part of our senior management to scale up the company in size and investment, so as to take on larger projects on a wider business scope. The financial assets available are enough for such ambitions, but “Mythical Man-Month” issues rule out the hiring of more people. The slow and unsteady progress of tasks is obvious. Even though the condition is very serious, it is not felt throughout the company; but, if not tackled, the situation might easily become extremely critical and unstable, and result in total collapse.

“Walking through a mine-field” has become our dilemma these days: We can keep going forward, as we have the necessary strength and means, but the consequences might be dire. We might even risk the loss of our essential assets.

In short, we do have our own share of the market, but Complex Dynamics is handicapped with its woeful software development process. We have invited many consultants to help us out, but to no effect. It seems that our "Gordian Knot" requires an Alexandrian solution: We must forget our defective software development process and engineer a new one. We therefore decided to seek the help of a methodology engineer (whom we will ironically call “Alexander”).

On his first visit, Alec declared that we do indeed have a serious problem. Since he needed a thorough survey of the company, we took a CT scan of the company: All the relevant personnel were interviewed and asked about the present situation, the history of the problems that they currently face, and their predictions for the future.

The accounts are reported hereinafter:

1- Staff-member X: When N teams at M sites are working on the same project simultaneously (i.e., in a distributed software development environment), effective collaboration, facilitated communication, a well-defined plan, and reliable scheduling are
essential. Without these, all endeavors are doomed to failure. No wonder we are in such trouble: We do not have any of the above.

2- X: The senior manager, the project manager, and the business manager, are all technocrats and capable people. That is a fact. But what I tell them is this: What about us? More precisely, what about our rights? We have the right to know about your decisions and agreements. What have you agreed on with the customer? Where is the contract? How long should the project take? By when should the first release be out? Which factors are most important for the customer? We have never seen the customer here. Obviously, if these issues are neglected, we would have an angry customer with a useless toy in hand, not a useful software system. We cannot build the system out of thin air. We could no longer sit tight and put up with these problems. We can complete the tasks if properly supplied with user feedback; but when a project fails due to lack of proper feedback, all the blame is put on us, the developers.

3- Process manager: X left the company just recently. I mean he himself went off in a huff. I do not know the reason. What difference does it make, anyway! What is important is that he is not here anymore. Of course, he took his knowledge with him, especially all the expertise that we had on Oracle. So the company huffed and puffed and set out to train another staff member, Y. After all, our most valuable capital is what we invest in our human resources.

4- Staff-member Y: This Oracle training was taking too long, and I was under a lot of pressure to finish it ASAP. The company could have easily foreseen that X would leave someday. People are leaving all the time. The company should prepare itself for such contingencies. Risk management has been thrown to the wind.

5- Staff-member Z: I have no time for this interview. Please pose your questions to my colleagues. I am very busy with my projects, and cannot add an extra ball to the ones that I am already juggling.

6- Staff-member W: Little in our process is based on a solid foundation or is acquired systematically, and we have no criteria or set of metrics to assess our process and products. Procedures for testing, design and implementation are not well-defined. Modeling is not taken seriously; complexities are handled through heavy collaboration and communication among team members. In the absence of essential documents, certain people have to play the role of a blackboard, even a black recorder box: “What was decided last week? What was our agreement? Who is responsible for what?” Even the decisions made at formal sessions are not documented properly.

7- Y: We do not prepare any architecture or architectural view during our projects. We do not have any verification activities either. Although we do have review sessions, I do not know why almost all the sessions held at our company fail to achieve useful results; the resulting decisions are mostly vague and ineffective, and the artifacts produced are useless. We often realize after each session that we have been wasting valuable time. We sometimes end up in a wrangle (corncobs are everywhere), or reach a compromise based
on "Design by committee": Everybody agrees with the result but the goal has been missed.

8- En masse: This is not fair. We have no idea where the project is headed. We do not have any plans or serious planning activities; although we do have a so-called "project plan", it is actually nothing but an ornamental artifact. If we were to follow it, we would fall behind schedule in a very short time. On the other hand, the plan is never updated because our contracts are based on our plans, and our contracts cannot be changed. Unfortunately, we are irrational in our projects.

9- Process manager: Nobody knows that methodologies other than RUP actually exist. Everybody is afflicted with severe model-phobia; they are all fed up with models and modeling. These misperceptions must be corrected.

10- Process manager: Some developers do know what a CRC Card is. But it is more important to know to whom the CRC Card is beneficial, and how it should be used. For team members, the rationality of methods and tools remains unclear, and they are never explained. As a result, team members think that they are dealing with ornamental entities which serve no clear purpose.

11- Process manager: At our sites, we have several teams but no teamwork; that is, what we currently have are not proper teams. The environment and the tools must be improved so that teamwork and team management are facilitated.

12- Senior manager: In today’s market, innovation and quality are the factors which give us our competitive edge. The newest innovations in software and hardware platforms, technologies, and supporting paradigms have to be immediately incorporated in and supported by our products. Our products continue to evolve and adapt to the advances in technology. The pursuit of innovation, quality and value is our constant endeavor.

13- Process manager: [referring to the senior manager’s words] Actually, these goals have never been achieved. Nevertheless, the goals are not impossible to achieve; other companies have been quite successful. We could too, provided that we improve our processes. Regretfully, our software products are not even maintainable, let alone “evolvable and adaptable”.

14- Senior manager: We have established ourselves as producers of various product families, including digital archiving systems, Knowledge management systems, and digital library solutions. All our products are web-based, so we should be able to benefit from continuous, distributed and mass user feedback. We must be able to process and exploit this valuable feedback through our software evolution process.

15- Process manager: Staff-member X rejoined the company later on. He was put in charge of digging out our lava flows (dead code). But the projects in which he had been involved had moved on, and he was not able to cope with the complexity. The senior manager asked us to help him out, but we had no program comprehension tools, and there was no effective, consistent and up-to-date documentation to rely on. The situation soon spiraled
out of control. Our processes must be improved so as to avoid such strenuous situations in the future.

16- Process manager: Staff-member X has left the company yet again, this time for graduate study! People are coming and going all the time. Expensively acquired Knowledge and skill – invested in our human resources – is easily lost.

17- Process manager: Because of the large number of incomplete and failed projects, all the team members are worried about their projects. We should be able to formally measure and confidently declare desirable progress in our projects.

18- Process manager: Distributed development and remote teams; these conditions have made seamless modeling an absolute necessity.

19- Senior manager: We often take projects greedily, hastily and without giving proper consideration to our capabilities and limitations. Many projects turn out not to be profitable or in accordance with our resources and skills.

20- En masse: Mistakes tend to recur. We do not have any procedure or method to learn from past projects. Mistakes are therefore repeated, and we have to constantly deal with the painful consequences.

21- Process manager: For a project to fail, all that is needed is one – just one – complex use case. Since we do not use adequate modeling notations and techniques, ambiguities, uncertainties, complexities, inconsistencies, and misconceptions continue to take a heavy toll.

22- Process manager: Our teams are issue-driven. Instead of avoiding and mitigating the risks, we recklessly confront them and waste our time on solving them one by one. Team energy is depleted fast, and performance suffers. What we need is proper planning and management.

23- Z: It seems that face-to-face management, knowledge acquisition and sharing are not working (at least not for our company). The working environment should encourage and support self-organization. It is not acceptable to be spending this much time on attending speeches and meetings.

Assignment Guidelines

To address the above problems, sum up your knowledge of “Software Development Methodologies”, and perform the following steps:

**S1: Project characterization**- Study the narration thoroughly and characterize the situation described. Project Characterization aims at identifying the features of the specific situation of the project at hand.
**S2: Identifying the methodology requirements** - In this activity, the following steps should be followed:

1. Translate the project characteristics specified in the previous steps into a set of methodology requirements.
2. Add the requirements which should be satisfied in any typical software development methodology.

   [You are familiar with some of the methodology requirements: Refer to the criteria introduced in Lecture 1]

**S3: Designing the target method** - In this activity, the following steps should be followed.

1. Choose an appropriate method development strategy from among the following approaches:
   a. Assembly-based,
   b. Paradigm-based, or
   c. Extension-based.
2. Use your current knowledge of methodology engineering approaches and the variety of software development methodologies with which you are familiar.

**S4: Implementing the target method** - In this activity, the following steps should be performed:

1. Add the required details to the designed method so that it is implementable in the situation.
2. Use Eclipse Process Framework Composer (EPFC) tool for implementing the methodology.

*Mohammad Reza Besharati, Ali Sepehri Khameneh*

*Fall 2012*