The previous set of notes left off with the concept of classifying requirements based on several type of criteria. This is the first step in analyzing and ultimately defining the requirements for a project (or portion of a project). We continue our discussion of requirements engineering by elaborating on the analysis of the collected requirements in Section 4. We follow that up with a discussion about the negotiation process in Section 5, investigating why we must negotiate requirements, what problems can occur, and what we can do to minimize or eliminate those problems.

4 Requirements Analysis

We analyze requirements for several reasons. Obviously, organizing the information is an immediate benefit, and the organizational structure we use draws on our own understanding of the problem domain, as well as on the basic type of requirements we defined in the previous section. The classification allows us to explore and develop the relationships between different information items. These relationships form the core of the requirements model.

We must ensure that the gathered requirements are consistent. This means that we do not have items that negate or contradict other items. For example, consider the following requirements for a system that automatically sends college application forms to prospective students based on their SAT scores or high school GPA:

- If a student’s combined SAT score is greater than 1400, the system shall send the forms; otherwise, the system shall send a rejection letter.
- If a student’s high school GPA is greater than 3.7, the system shall send the forms; otherwise, the system shall send a rejection letter.

The contradiction should be obvious to the reader. Each of these requirements mandates that the system reject an applicant that would be accepted by the other, yet part of both requirements are valid for all applicant.

We must also examine the set of requirements for omissions and ambiguity. Each requirement should clearly state exactly one need the completed system should meet. Requirements should be stated in language that is relevant to the problem domain and/or business model, and should have the same meaning to both the requirements engineer(s) and the customer(s).

Pressman[6] lists several questions that should be asked and answered when analyzing requirements:

- Is each requirement consistent with the overall objective for the system/product?
- Have all requirements been specified at the proper level of abstraction? That is, do some requirements provide a level of technical detail that is inappropriate at this stage?
- Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?
- Is each requirement bounded and unambiguous?
- Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement?
- Do any requirements conflict with other requirements?
• Is each requirement achievable in the technical environment that will house the system or product?
• Is each requirement testable, once implemented?

Unless you are very lucky (in which case you should be suspicious), you will discover problems with the gathered requirements. Usually this will involve conflicts or ambiguity that will generally require additional discussions with the customer.

5 Negotiating Requirements

Negotiation can be defined [1] as “a discussion intended to produce an agreement.” In one way or another, we will have to negotiate requirements with the customer. Negotiation is a social as well as a technical process. As most agile methodologies recognize, involving the customer throughout the development process is a critical aspect of building quality software. Boehm and Egyed note [2] that “its [requirements negotiation] results have probably the most significant effect on the system’s value.” Yet requirements negotiation can be one of the most difficult and frustrating activities in software engineering. Today’s technology has not made the task significantly easier, and has introduced some additional problems into the process.

It should be obvious that effective communication is essential to the negotiation process, and this requires a common understanding of the problem domain, with and without the system in place, as Zave and Jackson conclude in [8]. There are many factors that make this communication difficult. In [5], Herlea notes these key problems:

• The diversity of expertise and organizational roles impart different views and expectations of the system.
• Users are problem owners, consumers, and actual participants in the system. It is rarely easy to determine if they are formulating the right problem, or if they are describing the problem properly.
• Interpersonal and political conflicts within the organization can hamper open and honest communications.

To this list, we can add some additional problems that are a product of our distributed personnel and knowledge resources:

• Participants may work in geographically distant locations.
• Language itself may be a limiting factor as business concerns cross national boundaries.
• Participants may work for organizations outside the primary customer. These participants may have concerns and requirements unknown to the primary, or may have goals that adversarially compete with the primary's goals.

There are a wide variety of people who have an interest (we call them stakeholders) in the development of a software system. Obviously, the customers and users of the systems, and others who work in or with the environment where the system will be used. Less often considered are those who will design, build, install, and maintain the system, yet these people also have a vested interest in the product. All of these stakeholders have their own view of how the system will interact with the environment, and how they will interact with the system. Seen in this light, the requirements negotiation task can be viewed as a goal-oriented competitive process.

In this view, we say that each stakeholder has a set of goals that they expect the system to achieve. These goals are defined by their requirements. If a stakeholder strives to achieve a particular goal at the expense of others, we consider the competition to be adversarial, while if there is cooperation, the competition is cooperative. Robinson [7] uses the terms distributive and integrative, respectively,
to refer to these two models of competition or bargaining. While the cooperative model is a preferred basis for negotiation, we must also be aware of, and prepared for, adversarial competition.

In either mode of negotiation, the foundation must be laid by identifying the goals and objectives of all parties in commonly understood terms. Communication methods and protocols (e.g., teleconferences, face-to-face meetings, e-mail or newsgroup discussions, etc.) must be defined [5]. There must also be a change management system in place to record and manage documentation and other artifacts of the negotiation process [4].

The objective of the negotiation is to arrive at a set of requirements that satisfies all the stakeholders, while remaining complete, consistent, coherent, and testable. There will usually be compromises made, but these should involve user needs rather than violations of correct requirements. Compromises should balance risks, exposure, and estimated cost against simplicity and the benefits returned.

In many ways, we all negotiate on a daily basis. Sometimes, the negotiation is internal: how can I afford the new CD I want; should I stop for gas on the way to school, or wait until the trip home. Other times, we negotiate with others — where to eat lunch, when to meet to study, etc. There are some basic skills, techniques, and strategies that can help you become a more effective negotiator. We list ten of them here.

1. Know yourself. Before beginning a negotiation, understand how you feel about the process of negotiating. If you just want to “get it over with,” you may give away too much of your position. If you simply must be the winner in the negotiation, you run the risk of becoming adversarial and causing the negotiations to fail.

2. Know who you are negotiating with. In the requirements negotiation process, you will typically be working with more than one person, from different areas of the business. It can be very helpful to know the negotiating strategies and reputation of everyone involved, including your team mates.

3. Think ahead. Understand your position(s) on the point(s) to be negotiated. You should also be as informed as possible about the other positions represented and what their expectations are. To get ahead of the game, put yourself in the other’s position(s) and anticipate what they think of you and your position.

4. Build trust. Negotiation is a form of communication, and trust is essential. The other parties involved need to know that you will honor your commitments, and that you expect the same from them. Do not make promises you cannot keep. Always be honest about your positions.

5. Listen, listen, listen. Listening is one of the most important skills a good negotiator possesses. You should listen with your eyes as well as your ears. Watch facial expressions and body language for clues and nonverbal messages.

6. Be able to be flexible. Negotiations begin with positions or goals. As the communication builds and trust is established, the process should move to a discussion of needs and wants. If you’ve built a frank and honest line of communication, all parties will usually be willing to make compromises for the greater goal.

7. Maintain a balance of power. Do not assume that the others involved have the stronger position. You should enter the process confident in your own position(s) and with the knowledge to justify them. On the other hand, do not try to dominate the negotiations. This can lead to animosity and lead to an adversarial process. If you understand your position, you know where you can give and where you cannot. Remember that the balance of power will fluctuate over time.

8. Know you alternatives. What are your choices? What if an impasse is reached, and a particular point cannot be resolved? How will this affect the project, and what are the pros
and cons of those effects from your point of view? What about from the other parties’ viewpoints?

9. **Know when you’ve won.** What are your best-case and worst-case scenarios? The range between these cases is your *settlement range*. Define this range for yourself for each item under negotiation. Try to understand the settlement range of the other parties involved. A resolution in this range is a Win; ideally, all parties should see the resolution within their own settlement range.

10. **Have fun!** Negotiation is an ongoing process, not a single event. Understand the basic phases of the process: preparation; establishing the communication and building trust; identifying positions, interests, and needs; and agreeing upon an outcome.

These tips are, by design, fairly general, and can easily be applied to many different negotiation scenarios. However, they are all important to the requirements negotiation process. Negotiation is a skill that can only be learned through practice.

Agile development strategies integrate requirements negotiation deeply into the development process. Customer representatives with full decision-making authority work with developers in every aspect of building the system. Agile methodologies generally also build systems in small increments, limiting the scope and magnitude of the new requirements implemented in a single development iteration. They generally work from the most essential requirements down to the least needed. This allows the most critical functionality of a system to be implemented in the simplest possible framework, minimizing the possibility of problems caused by extraneous system components. Frequent (almost continuous) integration and testing allows developers to identify problematic requirements quickly and specifically. Agile methodologies also generally do not build “complete” requirements documents in advance of the design phase. Incremental requirements are gathered and documented, often informally, in conjunction with each development iteration or sprint. Of course, historical documentation must be maintained as a reference for future iterations. In essence, the requirements and specification documentation is built as the system is developed. Classical methodologies that rely on an extensive requirements phase to collect and identify all of the system requirements before beginning the design phase lack this flexibility.

Interest in the area of requirements negotiation has been growing. In particular, there has been work towards providing automated tools to assist in the negotiation process. These tools are primarily designed to assist with documentation of the process, tracking conflicts and changes, facilitating communication by providing a one-stop information source. Notable examples are the WinWin Spiral Process[3], Oz[7], and TeamWave Workplace[5].

**References**


