# Chapter 13: Improving predictions, products, processes and resources

1. We could investigate the role of increased understanding of the product in decreasing fault density by comparing projects that used the inspection process with those that didn't around the same time. Two such projects would most likely exhibit about the same level of understanding of the product group. Or, we could look at the level of experience of the project participants and compare projects with similar experience levels. The other factor, sloppy inspection and development activities, might result in lower fault densities if many of the faults in the product are not found. To test this, we would have to look at field error data, i.e. failures reported by customers after the products are operational.

2. The benefit of incorporating such assumptions into the model is that it results in more realistic and accurate models. However, the drawback is that it is very difficult to ensure that the assumptions are correct. To be confident in our assumptions, we have to base them on empirical evidence. That is, they must be supported by data from past projects.

3. A systems dynamics model can be built in which the two major variables are security and performance. Incorporated into the model would be a number of assumptions about how different factors affect and are affected by security and performance. For example, the model might include assumptions about how much it costs to implement added security features into the system, and how much it costs to develop the system with higher performance.

4. The danger is that the SEL experience does not apply to your organization. If your organization differs from the SEL in terms of application domain, computing environment, typical project budget and schedule, or a variety of other factors, then you may not be as successful, or you may be more successful, at implementing Cleanroom. This is why a strategy of careful study and experimentation is important in deciding if a particular technology is well suited to a particular environment. Although the specific experiences the SEL had with Cleanroom might not be universally applicable, you could use the strategy that the SEL used to investigate Cleanroom in your organization.

5. The first use of a new technology such as rapid prototyping must be done very carefully. If possible, it should have been tried previously on a small, non-critical project. Ideally, it could be tried first on a parallel project, i.e. a project that is implemented twice, once with rapid prototyping and once without. In any case, its first few uses must be carefully monitored, both to ensure that rapid prototyping is actually being carried out (we cannot say we are evaluating the rapid prototyping process if that is not really what was being followed) and to find out what effect it is having, both positive and negative. This monitoring would be done via collection and analysis of data, including effort data (to make sure the rapid prototyping process is not actually requiring more effort than expected), schedule data (to make sure that it is not causing delays), and defect data (to make sure that the quality of the product is not suffering). The *post-mortem* analysis is particularly important in a project in which a new technology is introduced. It should concentrate on how the process was affected, both positively and negatively, by the introduction of rapid prototyping, how developers liked the new technique, if the technique was implemented correctly, and, most importantly, how it could be improved when/if it is used next.

6. One strategy is to compare data from projects before and after the certification, with respect to the following measures:

   - effort, normalized by size of project
   - calendar time, normalized by size of project
   - product quality
   - accuracy of estimation
   - revenue

   Also, the cost of the certification process itself must be measured. This includes the effort to implement process improvement practices, collect and analyze data, create and modify documentation, and perform assessments.

7. Some issues to consider:

    - What development activities should a licensed software engineer be able to demonstrate proficiency in?
    - If a software component, written by a licensed software engineer, fails with catastrophic consequences, should that engineer's license be revoked?
    - What should be the minimal educational requirements be for licensing (BS? MS? in computer science? in a related field?)?
    - How can we ensure that the criteria for licensing do not become outdated as software engineering technology evolves?
    - Should a licensed software engineer be able to continually improve his/her own skills, as we expect of development organizations?