Tips

1. Distinguish between estimates, targets, and commitments.
2. When you're asked to provide an estimate, determine whether you're supposed to be estimating or figuring out how to hit a target.
3. When you see a single-point "estimate," as if the number is an estimate or if it's really a target.
4. When you see a single-point estimate, that number's probability is not 100%. Ask what the probablity of that number is.
5. Don't provide "percentage confident" estimates (especially "90% confident") unless you have a quantitatively derived basis for doing so.
6. Avoid using artificially narrow ranges. Be sure the ranges you use in your estimates don't misrepresent your confidence in your estimates.
7. If you are feeling pressure to make your ranges narrower, verify that the pressure actually is coming from an external source and not from yourself.
8. Don't intentionally underestimate. The penalty for underestimation is more severe than the penalty for overestimation. Address concerns about overestimation through planning and control, not by biasing your estimates.
9. Recongnize a mismatch between a project's business target and a project's estimate for what it is: valuable rist information that the project might not be successful. Take corrective actions early, when it can do some good. The possible corrective actions are:
    - Redefine scope of the project.
    - Increase staff. Or, transfer best staff onto the project.
    - Stagger the delivery of different functionality. Or decide that the project is not worth doing after all.

Takeaways

- The primary purpose of software estimation is not to predict a project's outcome; it is to determine if a project's targets are realistic enough to allow the project to be controlled to meet them.
- If you took a quiz with ten questions and you answered each question with 90% confidence, your chance of getting all ten correct is 34.9%. Your chance of getting nine of them correct is 38.7%. You have a chance of 93% for getting at least eight correct.
- Most people's intuitive sense of "90% confident" ≈ "30% confident."
- Developers typically estimate 20% to 30% lower than their actual effort.
- The larger a project, in terms of LOC, the less chance the project has of completing on time and the greater chance it has of failing outright.
- Software industry has an underestimation problem. Before we can make our estimates more accurate, we need to start making the estimates bigger.
- Good estimates facilitate progress tracking through comparing planned progress against actual progress.
- Projects that aim from the beginning to have the lowest number of defects usually also have the shortest schedules.
- A project team that holds its ground and insists on an accurate estimate will improve its credibility within its organization.