

The background of the image features several bright blue, glowing streaks that originate from the right side and extend towards the left, creating a sense of dynamic movement and energy. The streaks vary in thickness and intensity, with some appearing as sharp lines and others as softer, more diffuse bands of light. The overall color palette is dominated by deep blues and blacks, with the bright blue of the light streaks providing a strong contrast.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*in the Name of Allah, the Beneficent, the Merciful*

# Software Quality Assurance

## Critical Estimation Concepts

# Estimates, Targets and Commitments

- *????*
- *????*
- *????*

# Estimates, Targets and Commitments

An Estimate is a prediction of how long a project will take and how much will it cost.

A target is a statement of desirable business objective  
*“we need to have version 2.1 ready in May, 2009 to demonstrate in trade show “*

A commitment can be the same as an estimate or it can be more aggressive or conservative than the estimate. Commitment and estimates can be different.



Tip “***Differentiate between estimate, targets and commitments***”



# Relationship between estimates and plans

Estimation is not planning and planning is not estimation. The goal of estimation is accuracy, the goal is not to seek particular result, but the goal of planning is to achieve a particular result. We bias our plans deliberately to achieve specific outcomes.

# Communicating about estimates, targets and commitments

## Project stakeholders sometimes miscommunicate about these activities

*EXECUTIVE: How long do you think this project will take? We need to have this software ready in 3 months for a trade show. I can't give you any more team members, so you'll have to do the work with your current staff. Here's a list of the features we'll need.*

*PROJECT LEAD: OK, let me crunch some numbers, and get back to you.*

*Later...*

*PROJECT LEAD: We've estimated the project will take 5 months.*

*EXECUTIVE: Five months!? Didn't you hear me? I said we needed to have this software ready in 3 months for a trade show!*

# Communicating about estimates, targets and commitments

From the previous example both the executive and project lead will walk away considering that the others are irrational.

The business executive was asking the team lead to come up with a plan that hits the target, more productive of the interaction could be as on next slide.



# Communicating about estimates, targets and commitments

*EXECUTIVE: How long do you think this project will take? We need to have this software ready in 3 months for a trade show. I can't give you any more team members, so you'll have to do the work with your current staff. Here's a list of the features we'll need.*

*PROJECT LEAD: Let me make sure I understand what you're asking for. Is it more important for us to deliver 100% of these features, or is it more important to have something ready for the trade show?*

*EXECUTIVE: We have to have something ready for the trade show. We'd like to have 100% of those features if possible.*

*PROJECT LEAD: I want to be sure I follow through on your priorities as best I can. If it turns out that we can't deliver 100% of the features by the trade show, should we be ready to ship what we've got at trade show time, or should we plan to slip the ship date beyond the trade show?*

*EXECUTIVE: We have to have something for the trade show, so if push comes to shove, we have to ship something, even if it isn't 100% of what we want.*

*PROJECT LEAD: OK, I'll come up with a plan for delivering as many features as we can in the next 3 months.*

**Tip** *When you are asked to provide an estimate, determine whether you are supposed to be estimating or figuring out how to hit a target.*

# Estimates as probability statements

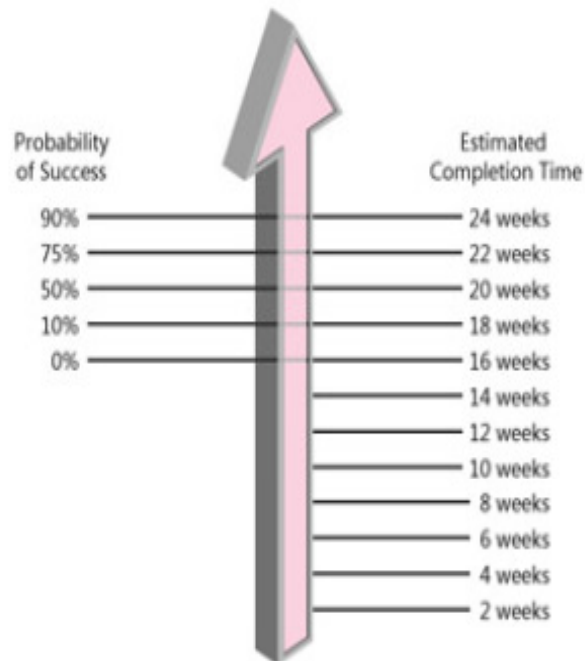


**Figure 1-1:** Single-point estimates assume 100% probability of the actual outcome equaling the planned outcome. This isn't realistic.

For Example: The project will take 14 weeks.

Tip ***When you see a single point estimate, ask whether the number is an Estimate or whether it's really a target***

# Estimates as probability statements

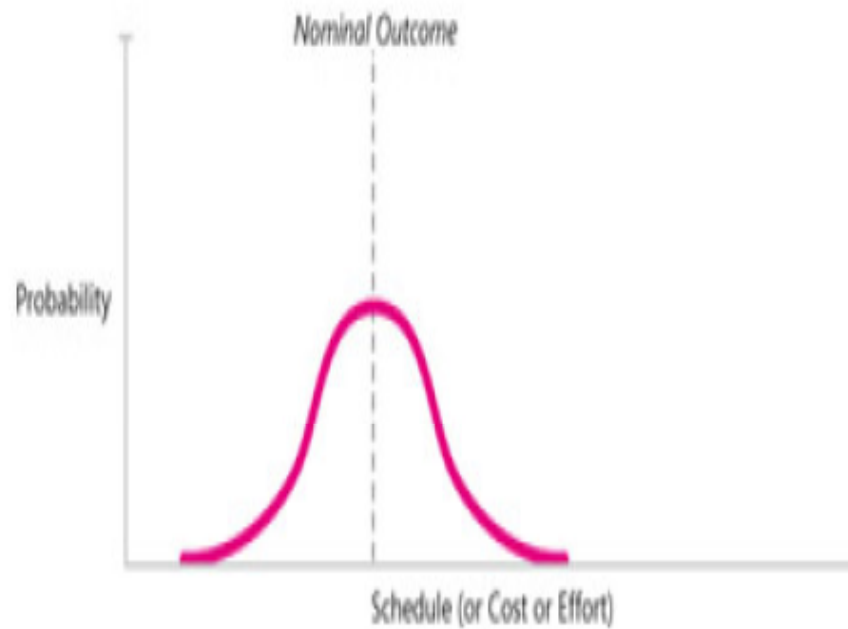


**Figure 1-5:** All single-point estimates are associated with a probability, explicitly or implicitly.

Figure 1-5 presents the idea of probabilistic project outcomes in another way. As you can see from the figure, a naked estimate like "18 weeks" leaves out the interesting information that 18 weeks is only 10% likely. An estimate like "18 to 24 weeks" is more informative and conveys useful information about the likely range of project outcomes.

**Tip** *When you see a single point estimate, that number's probability is not 100%. Ask what the probability of that number is.*

# Estimates as probability statements

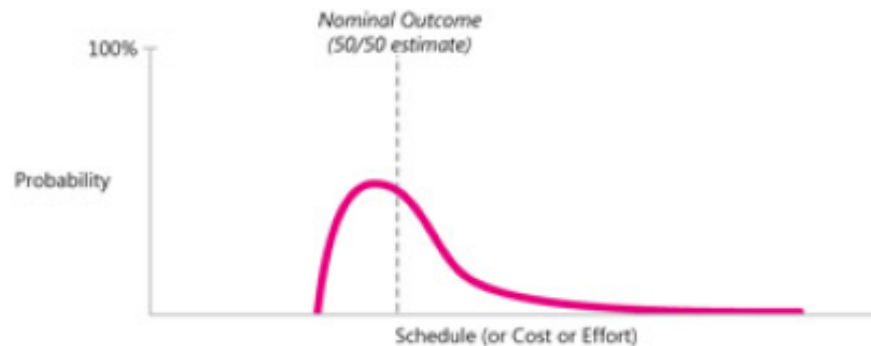


**Figure 1-2:** A common assumption is that software project outcomes follow a bell curve. This assumption is incorrect because there are limits to how efficiently a project team can complete any given amount of work.



# Estimates as probability statements

In reality there is a limit on how well a project can go but no limit on how poorly a project can go, so there should be a long tail on right side of the probability distribution.



**Figure 1-3:** An accurate depiction of possible software project outcomes. There is a limit to how well a project can go but no limit to how many problems can occur.

The vertical dashed line shows the "nominal" outcome, which is also the "50/50" outcome—there's a 50% chance that the project will finish better and a 50% chance that it will finish worse. Statistically, this is known as the "median" outcome.

# Common definitions of a good estimate

“Accuracy with plus minus 10% is possible, but only on well controlled projects”

“Good estimation approach should provide estimates that are with 25% of the actual results 75% of the time”

# Working Definition of a “Good Estimate”

“A good estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its target”

# Is it better to overestimate or underestimate

Accurate estimates are rare, so if we are going to err, is it better to err on the side of overestimation or underestimation?

# Arguments against over estimation

**If project is over estimated , Parkinson's Law will kick in;**

**Parkinson's Law is the based on the idea that work will expand to fill the available gap;**

**Most managers consciously squeeze the estimates to try to avoid Parkinson's Law.**



# What Managers do

*The developers say that this project will take 6 months. I think there's some padding in their estimates and some fat that can be squeezed out of them. In addition, I'd like to have some schedule urgency on this project to force prioritizations among features. So I'm going to insist on a 3-month schedule. I don't really believe the project can be completed in 3 months, but that's what I'm going to present to the developers. If I'm right, the developers might deliver in 4 or 5 months. Worst case, the developers will deliver in the 6 months they originally estimated.*

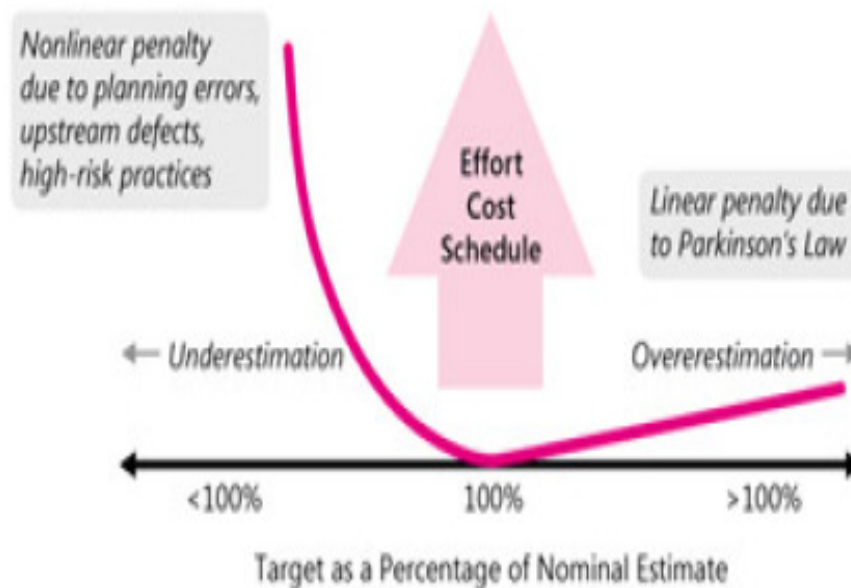
Are these arguments compelling? To determine that, we need to examine the arguments in favor of erring on the side of overestimation.

# Arguments against underestimation

Underestimation creates numerous problems:

- ✓ Reduces effectiveness of project plans
- ✓ Statistically reduced chance of on time completion
- ✓ Poor technical foundation leads to worse than nominal results
- ✓ Destructive late project dynamics make the project worst than nominal

# Weighing the Arguments



**Figure 3-1:** The penalties for underestimation are more severe than the penalties for overestimation, so, if you can't estimate with complete accuracy, try to err on the side of overestimation rather than underestimation.

**Tip** *Don't intentionally underestimate; the penalty of underestimation is more severe than overestimation. Address concerns about overestimation through planning and control, not by biasing your estimates*

## Benefits of accurate estimates

- ✓ Improved status visibility
- ✓ Higher Quality
- ✓ Better coordination with non software functions
- ✓ Better Budgeting
- ✓ Increased credibility for the development team
- ✓ Early Risk Information

**Tip *Recognize a mismatch between a project's business target and objective for what it is: valuable risk information that the project might not be successful . Take corrective action early, when it can do some good.***



# Value of predictability compared with other desirable project outcomes

Some of the goals, software strive for:

- ✓ Schedule
- ✓ Cost
- ✓ Functionality

What is important, the ability to change your mind about features, or the ability to know cost, schedule and functionality in advance? Majority project managers will opt for the later option. Everyone wants predictability.