
بسم الله الرحمن الرحيم
in the Name of Allah, the Beneficient, the Merciful

Software Quality Assurance

Critical Estimation Concepts

Four Generic sources of Estimation Error

- ✓ Inaccurate information about project being estimated.
- ✓ Inaccurate information about organization that will perform the project.
- ✓ Too much chaos in project to support accurate estimation.
- ✓ Inaccuracies arising from the estimation process itself

The Cone of Uncertainty

The cone shows how estimates become more accurate
As project progresses.

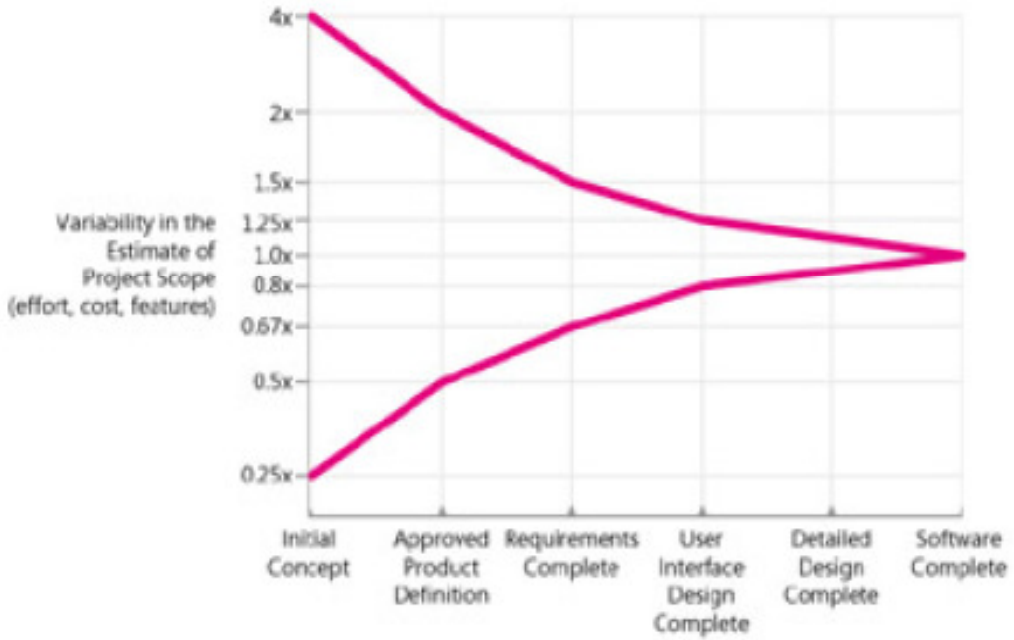

Figure 4-1: The Cone of Uncertainty based on common project milestones.

The Cone of Uncertainty

- ✓ The reason that estimate contains variability is that the software project contains variability.
- ✓ The only way to reduce variability in the estimate is to reduce the variability in the Project.
- ✓ One misleading implication is that the cone takes forever to narrow.
- ✓ See the diagram on the next slide

The cone of Uncertainty

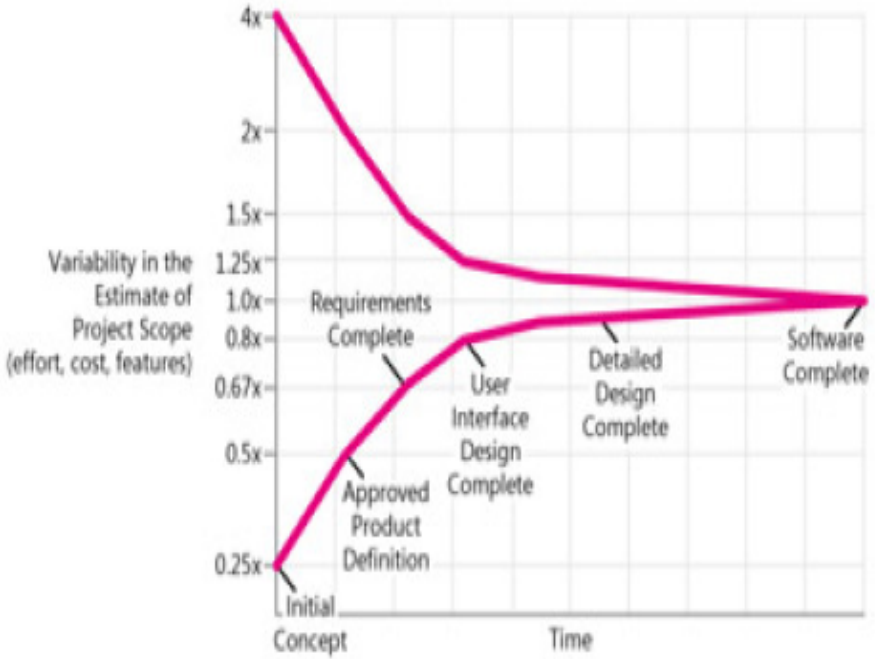

Figure 4-2: The Cone of Uncertainty based on calendar time. The Cone narrows much more quickly than would appear from the previous depiction in Figure 4-1.

Tip Regarding the cone

Tip: ***Consider the effect of the cone of uncertainty on the accuracy of your estimate. Your estimate cannot have more accuracy than is possible at your project's current position within the cone***.

The Cone doesn't narrow itself

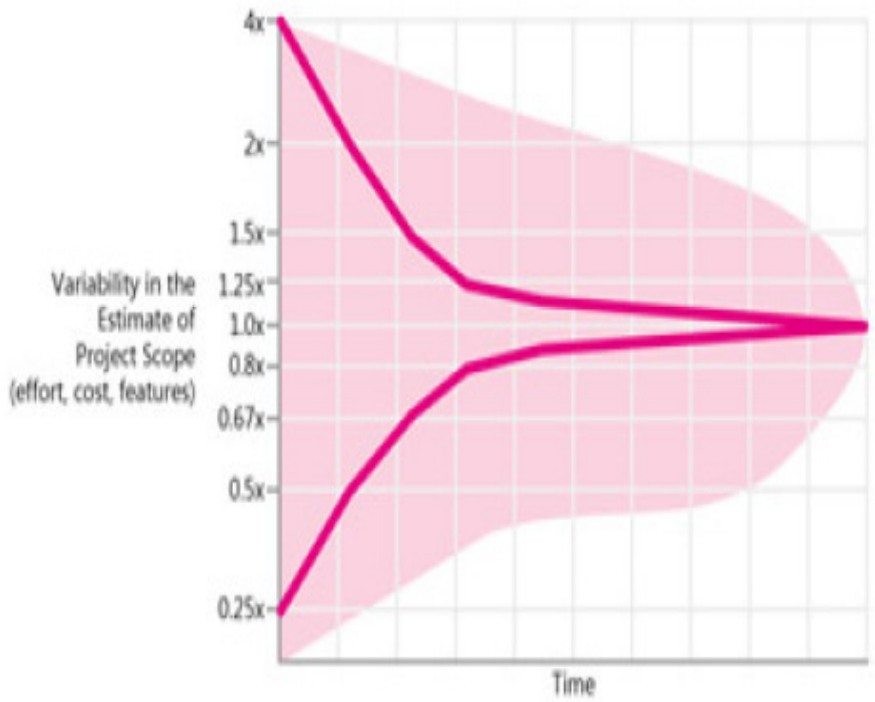

Figure 4-3: If a project is not well controlled or well estimated, you can end up with a Cloud of Uncertainty that contains even more estimation error than that represented by the Cone.

The cone doesn’t narrow itself

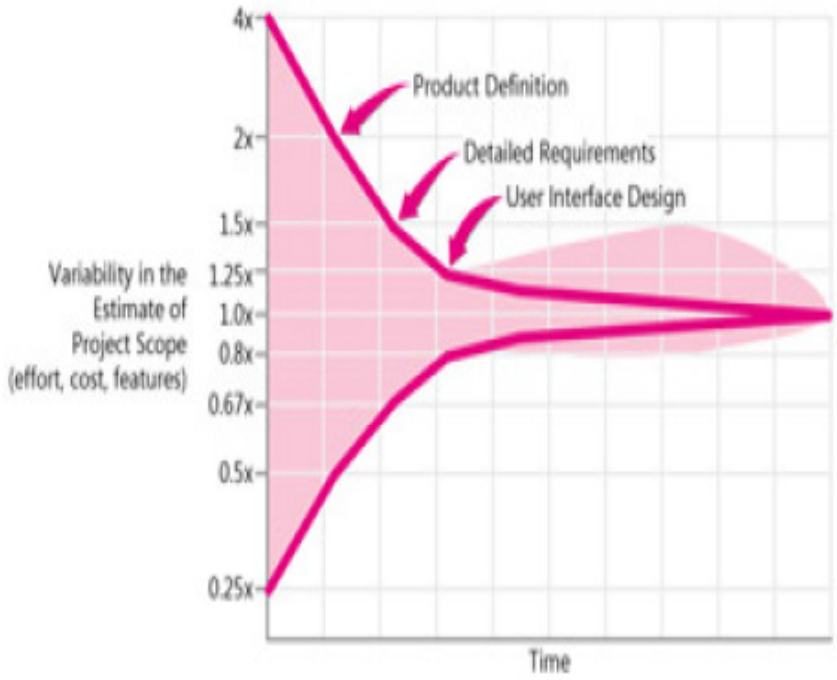

Figure 4-4: The Cone of Uncertainty doesn't narrow itself. You narrow the Cone by making decisions that remove sources of variability from the project. Some of these decisions are about what the project will deliver; some are about what the project will *not* deliver. If these decisions change later, the Cone will widen.

Another Tip

Tip: ***Don't assume that the cone of uncertainty will narrow itself. You must force the cone to narrow by removing the sources of variability from your project***

Two important tips

Tip: ***Account for cone of uncertainty by using predefined uncertainty ranges in your estimates***

Tip: ***Account for cone of uncertainty by having one person create the "how much" part of the estimate and a different person create the "how certain" part of the estimate***

Relationship between cone of uncertainty and commitment

Effective organizations delay their commitments until they have done the work to force the cone to narrow. Meaningful commitments in the early-middle part of the project (about 30% of the way in) are possible and appropriate.

Chaotic Development Processes

The cone represents uncertainty that is inherent even in well-run projects. Additional variability can arise from poorly run projects- that is from avoidable chaos
Common examples of chaos are:

- ✓Requirements that were not investigated
- ✓Lack of end user involvement in requirements validation
- ✓Poor designs that lead to errors in code
- ✓Poor coding practices
- ✓Inexperienced personnel
- ✓Incomplete or unskilled project management
- ✓Prima Donna team members
- ✓Abandoning planning under pressure
- ✓Developer gold-plating
- ✓Lack of automated source code control

Tip for chaotic process

Tip: ***Don't expect better estimation process alone to provide more accurate estimates for chaotic projects. You cant accurately estimate an out-of-control process. As a first step, fixing the chaos is more important than improving the estimates.***

Unstable Requirements

Requirement changes has often been reported as a common source of estimation problem. Unstable requirements present two specific estimation challenges.

- ✓ The first challenge is that unstable requirements represent one specific flavor of project chaos. If requirements cannot be stabilized, the cone of uncertainty can't be narrowed and estimation variability will remain high till the end of the project.
- ✓ The second challenge is that requirements changes are often not tracked and the project is often not re estimated when it should be

Tip for Unstable Requirements

Tip: ***To deal with unstable requirements, consider project control strategies instead of or in addition to estimation strategies.***

Requirements Growth

In order to estimate the effect of unstable requirements, you might consider simply incorporating an allowance for requirements growth, requirements changes or both in your estimate. A cone of uncertainty that accounts for requirements growth will also be different than the previous cone.

Omitted Activities

- This is an example of error arising from the estimation practices.
- We forget to include necessary task in the project estimates.
- Studies show that developers overlook 20% to 30% of necessary task that brings about a 20% to 30% estimation error.

Omitted Activities

Omitted works falls into three major categories:

- ✓ Missing Requirements
- ✓ Missing Software Development activities
- ✓ Missing non software development activities

Requirements generally missed

Table 4-2: Functional and Nonfunctional Requirements Commonly Missing from Software Estimates

Functional Requirements Areas	Nonfunctional Requirements
Setup/installation program	Accuracy
Data conversion utility	Interoperability
Glue code needed to use third-party or open-source software	Modifiability
	Performance
Help system	Portability
Deployment mode	Reliability
Interfaces with external systems	Responsiveness
	Reusability
	Scalability
	Security
	Survivability
	Usability

Tip for omitted requirements

Tip: ***Include time in your estimates for all requirements. Nothing could be built for free and your estimates should not imply that it can.***

Common software development activities missing from estimates

- Ramp up time for new team members
- Mentoring of new team members
- Installation
- Customization
- Participation in review
- Coordination with stake holders
- Defect correction
- Maintaining the revision control system
- Coordinating with sub contractors

Tip for missing software development activities

Tip: ***Include all necessary software development activities in your estimates, not just coding and testing.***

Non software development activities commonly missed

Table 4-4: Non-Software-Development Activities Commonly Missing from Software Estimates

Vacations	Company meetings
Holidays	Department meetings
Sick days	Setting up new workstations
Training	Installing new versions of tools on workstations
Weekends	Troubleshooting hardware and software problems

Some projects deliberately plan to exclude many of the activities in Table 4-4 for a small project. That can work for a short time, but these activities tend to creep back into any project that lasts longer than a few weeks.

Tip for missing non software development activities

Tip: ***On projects that last longer than a few weeks, include allowances for overhead activities such as vacations, sick days, training days and company meetings.***

Unfounded Optimism

Microsoft Vice President Christ Peter observed that "***You never have to fear that estimates created by developer will be too pessimistic, because developers will always generate a too optimistic schedule***"

Tip: ***Don't reduce developer's estimate- they are probably too optimistic already.***

Unfounded Optimism

Common reasons for being optimistic are:

- ✓ We will be more productive on this project than we were on the last project.
- ✓ A lot of things went wrong on the last project. Not so many things will go wrong for this project.
- ✓ We have learned a lot of lessons that will allow us to finish this project much faster.

Subjectivity and Bias

Subjectivity creeps into estimates in the form of optimism, conscious bias and unconscious bias.

Estimation Bias: Intent to fudge an estimate in one direction or another

Estimation Subjectivity: Human judgment is influenced by human experience, both consciously and unconsciously.

Subjectivity and Bias

Bias occurs when estimates are not aligned with the targets and pressure is applied to the estimate, project and project team. It is said that excessive schedule pressure occurs in 75% to 100% of large projects.

Subjectivity occurs when we feel when considering different estimation techniques that the more control knobs we have the better estimate we will get. The reality is opposite the more control knobs we have the more chances of subjectivity to creep in.

Subjectivity improves if we have less control knobs

In contrast, Figure 4-7 illustrates the range of estimation outcomes with an estimation technique that includes only one place to insert a subjective judgment into the estimate—that is, one control knob. (The control knob in this case is unrelated to the Cocomo II factors.)

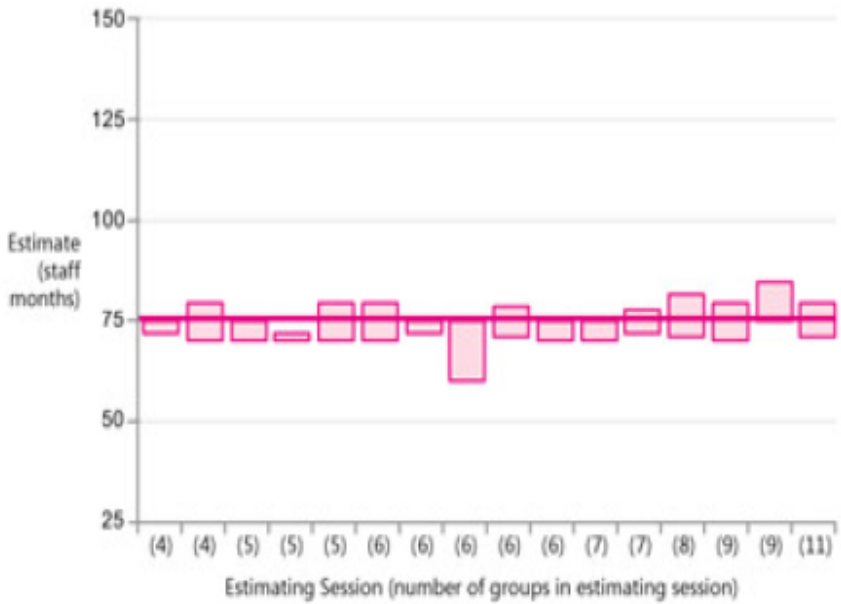

Figure 4-7: Example of low variation in estimates resulting from a small number of adjustment factors. (The scales of the two graphs are different, but they are directly comparable when you account for the difference in the average values on the two graphs.)

Tip for subjectivity

Tip: ***Avoid having "control knobs" on your estimates. While control knobs might give you a feeling of accuracy, they usually introduce subjectivity and degrade actual accuracy.***

Off-The-Cuff Estimates

Intuition and guess in software cost estimates are both correlated with cost and schedule overruns.

Tip: ***Don't give off-the-cuff estimates. Even a 15 minutes estimate will be more accurate.***

Other sources of error

Some other sources of estimation error are:

- ✓ Unfamiliar business area
- ✓ Unfamiliar technology area
- ✓ Incorrect conversion from estimated time to project time