Estimation Checklist

Requirements: Make sure all the requirements are clearly spelled out.

Resources & Environment: Understand the environment and resources you have to complete the work.

Books: Steve McConnell – Software Estimation Demystifying the Black Art.

If You Don’t Know: Say “I’ll get back to you.”

How Critical Is Estimate: The more critical and larger the project, the more time you should spend estimating.

QA: Did you factor in QA time.

Critical Thinking: Ask questions surrounding Methodic Process Checklist. What are we going to build, how, who, how fast.

Cone of Uncertainty: The earlier you are in a project the more you are likely to miss your estimate.

C:\Users\Admin\Downloads\IMG_1698.JPG

Things to help reduce estimate uncertainty: Approving product definition, clear and concise requirements, user interface design complete, detailed design complete.

Spike: Taking time to identify requirements analysis.

Don’t Overestimate: If you pad, pad in just one place.

Day isn’t a day: Not all developers get 100% day due to other responsibilities, vacation, support, etc.

Ramp Up: Make sure to consider ramp up time.

Documentation: Make sure to consider handoff and documentation.

Take into consideration team size: The more developers assigned to a project, the more communication and the less effective each developer will be.

Direct Costs: Cost of time for developers, testers.

Indirect Costs: Cost of time for project managers, architects.

Ways to estimate: By story, count of components, relative to other similar projects, blind bidding.

Blind Bidding: Multiple people each take a guess and reveal their estimate at the same time.

Story Buckets: Estimate in terms of half day, day, half week, week. If something is greater than a week, split it up into smaller components.

Week Sprints: Week sprints are recommended because you are never a week away from knowing that there is a problem.

Law of Large Numbers: One large estimate could be off by a large number in either way. Many small estimates tend to average out.

Things that often blow up an estimate: UI, missing requirements, extra time to standup scaffolding, implementing customer feedback from acceptance testing.

Testing: Estimates should include testing time.