

Best Practices for Software Projects - Estimating

April 2004 - Pragmatic Software Newsletters

Best Practices for Software Projects - Estimating

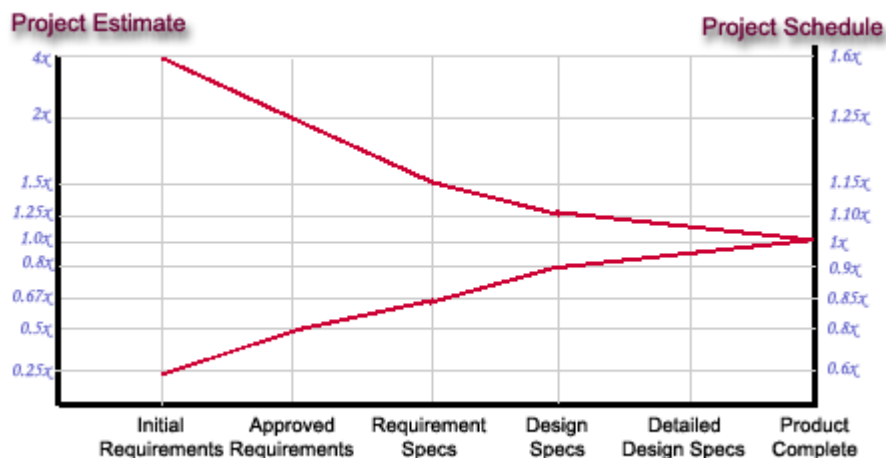
Ever have a manager give you a brief description of a possible project and ask you for an estimate? After listening to the manager describe the project, you give him a "seat-of-the-pants" estimate, explaining that that it is a ball-park estimate based on what you've been told.

To your surprise, you learn a week later that the manager has presented the estimate to a client and puts you in charge of delivering a complete solution based on the estimate you provided. You shockingly explain that the estimate was just a ball-park estimate and you need to learn the detailed requirements before you can provide a final estimate. The manager argues that it is too late for that, saying that the customer has already approved your original estimate and the manager expects you deliver what you initially agreed to.

Reluctantly, you proceed with the original estimates. 12 months later, your manager calls you into his office and asks why you are already 4 months late delivering the project and wants to know how you could be so incompetent in your estimating (not acknowledging that it was his fault for communicating the ball-park estimate to the client in the first place)! Does this sound familiar?

Unfortunately, this happens often. Most projects are underestimated by anywhere from 25 to 100 percent, and this is normally caused by providing preliminary estimates that are published as the "golden estimate". To provide a better estimate, you must make it all the way through the product design phase before publishing the estimate, where it will have an error margin of 25%.

Researchers have found that project estimates fall within a predictable range at various stages of the software lifecycle. According to Boehm 1995 (Cost Models for Future Life Cycle Processes COCOMO 2.0), your estimates will improve as the software lifecycle matures:



As you can see, it is practically impossible to provide an adequate estimate in the Initial Requirements stage, so don't fall into that trap. Below are some tips for providing better estimates:

- **Management Education** - If your manager is not an experienced project manager, explain the estimating process to your manager so that they understand how estimates improve as you know more about the project.
- **Document Initial Estimates as Ball-Parks** - If your manager requires an initial estimate, submit your estimate to them in writing, explaining in detail that the estimate is a ball-park and should not

be taken as a reliable estimate at this stage of the project.

- **Detailed Design Specs** - Insist that your team meets with the client to hash out all features of the product. Then take those requirements back to your technical team to get detailed design specifications. If the detailed design specifications are done properly, your margin of error for the estimate should be less than 25%.
- **Risk Mitigation** - Document your project risks and add additional hours to your project plan for risk mitigation.
- **Project Management** - To stay on track with your deliverables, your team should meet regularly to review progress and to aid team members that are falling behind. Team leaders should provide regular code reviews to shorten testing cycles and to prevent major errors from stalling your project schedule.
- **Online Collaboration** - Use an online collaboration tool (web-based preferably) to manage your entire life cycle and your project tasks as to improve project collaboration and communication. A solid tool for this is Software Planner (<http://www.softwareplanner.com>).

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines** - <http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf>
- **Functional Specifications** - <http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf>
- **Architectural Overview** - <http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf>
- **Detailed Design** - <http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf>
- **Strategic Planning Document** - <http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf>
- **Test Design** - <http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf>
- **Risk Assessment** - <http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf>
- **Weekly Status** - <http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf>
- **User Acceptance Test Release Report** - <http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf>
- **Post Mortem Report** - <http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf>
- **All Templates** - <http://www.PragmaticSW.com/Templates.htm>
- **Prior Newsletters** - <http://www.PragmaticSW.com/Newsletters.htm>
- **Software Planner** - <http://www.SoftwarePlanner.com>

About the Author

Steve Miller is the President of **Pragmatic Software** (<http://www.PragmaticSW.com>). With over 20 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at <http://www.PragmaticSW.com/Newsletters.htm>. Steve's email is steve.miller@PragmaticSW.com.

Pragmatic Software Co., Inc.
1745 Shea Center Drive
Suite 400
Highlands Ranch, CO 80129

Phone: 720.344.4846
Fax: 720.344.4847
Web site: <http://www.PragmaticSW.com>
E-mail: info@PragmaticSW.com