White Paper

Software Estimation - Inside Out

Tabinda Aftab

Ephlux

White Paper

Abstract

Why do software projects do not meet the deadline? Why do they frequently overrun on cost more than originally planned? In a recent survey, 40% of companies reported failed software schedule and budget estimation while only 14% reported good performance. What is it that the 14% do right and what is it that the 40% do wrong?

Just as typically every real world task needs to determine the dimensions, scope and features/characteristics before implementing the task similarly software planning requires sizing and scoping a project as practically as possible. It is generally believed that software sizing is difficult and fail prone as compared to other real world projects such as construction or engineering related projects , the main reason is that enough history and facts are not available that could be used as benchmark.

Estimating a project incorrectly by either estimating the size too low or not providing required resources or time will result in a failed project. Metrics gives the ability to identify, resolve, and/or limit risk before they appear.

Software size estimation is one of the most critical and difficult activity in the software planning process. Measuring the software size goes a long way in determining efforts, schedule and budget for a project. Any failures in misjudging or misunderstanding the scope and size of the project will always result in over budget, failed deliveries and schedule overruns thus resulting in total failure of the project.

Software size can be measured on several parameters such as lines-of-source code, Function Points or Use Cases and Feature sets. Each parameter has its own scenarios, advantages and limitations. No matter what parameter is used for estimation the account for project complexity is mandatory in identifying the actual efforts , resource and time estimates. Another very important factor is estimating project size is the identification of scope and requirements. Requirements creeping in later stages of project misbalance the estimates and thus affect all the sizing parameters. Continual use of these measurement metrics will assist any organization in identifying their organizational trends as market facts and trends can be misleading for different companies. A measurement history will in turn assist to better estimate and plan a project.

Why Estimates Fail?

There are various reasons for estimation failures and inaccuracy. Estimation requires not only facts and data but also needs a positive frame of mind to accept and identify every ambiguity and difficulty. Following are some of the common practices that result in ineffective estimates thus resulting in failed and off track projects:

Over Confidence: Management and business managers involved in estimation and planning generally make estimates considering best case scenario. No possible wrong turns or problems are identified or considered while the reality is opposite to this. Thus the estimates of time and budget disturb just as the first problem is encountered.

Anchoring: When difficult situation arises during planning and estimation many people tend to make a "good guess" that will anchor their planning and thinking all along the process. The involved team grasps this likely or unlikely answer and adjusts it to match their requirement without considering scenarios where this guess will fail. Anchoring stops the team to include possible worst cases that would have changed the estimates otherwise. Many times these anchors can be feed into through customer requirements such as "The customer says he needs the project in 2 months".

Team Decision: Even though team work is a good exercise but sometimes while estimating and sizing a project a popular idea within the team can rule out a minority thought only because most of the people agree to the popular idea. This behavior can sometimes ignore potential scenarios or feature/function thus resulting in inaccurate estimates that might get known not until very later in the project lifecycle.

Hindsight: Many times people believe that the outcomes associated with the project are very predictable and do not consider any twists or turns at all.

Management Behavior: In several organizations the planning and estimation activities are governed by the management's behavior that limits some important parameters and thus accurate estimation cannot be done.

Selective Perception: The team perceives only what was expected to perceive and disregard any other conflicting detail. Many times the teams/estimators are not able to distinguish between 50%, 75% and 90% probability of their perceived thoughts when scoping, defining and sizing.

Classification &breakdown: In order to estimate the resource efforts and timelines on a project accurate functional breakdown and classification of functional aspects of the project are mandatory.

Software Sizing Techniques

There are three basic methods for measuring software size. These techniques have their own pros and cons. Following is a brief about each technique:

Source Lines-of-Code Estimates

Most SLOC estimates count all executable instructions and data declarations but exclude comments, blanks, and continuation lines. SLOC can be used to estimate size through analogy by comparing the new software's functionality to similar functionality found in other historic applications.

The most significant advantage of SLOC estimates is that they directly relate to the software to be built. The software can then be measured after completion and compared with your initial estimates.

However, it is virtually impossible to estimate SLOC from initial requirements statements. Their use in estimation requires a level of detail that is hard to achieve (i.e., the planner must often estimate the SLOC to be produced before sufficient detail is available to accurately do so.)

Because SLOCs are language-specific, the definition of how SLOCs are counted has been troublesome to standardize. This makes comparisons of size estimates between applications written in different programming languages difficult even though conversion factors are available.

Function Point/Use Case Estimates

Function points are the weighted sums of five different factors that relate to user requirements:

- Inputs,
- Outputs,
- Logic (or master) files,
- Inquiries, and
- Interfaces

A function point or use case can be defined as a low level /high level task that the system is expected to perform. During estimation the entire system shall be breakdown into smaller user cases to determine the scope and size of the system.

Once these use cases are identified resource, effort and timelines can be attached to each use case.

The sum of the total complexity-adjusted function points/use cases (for all types of function points) becomes the total adjusted function point/use case count. Based on prior experience, the final function point figure can be converted into a reasonably good estimate of required development resources.

While function points aid software size estimates, they too have drawbacks. At the very early stages of system development, function points are also difficult to estimate. Additionally, the complexity factors applied to the equation are subjective since they are based on the analyst/engineer's judgment. Basically, requirements are rarely fully gathered up front for anything but the most trivial projects. You want to gather as much as you can, but an iterative process of gathering requirements and building from those requirements will ensure that the project runs as smoothly as possible and you will have a documented trail of the decisions made should something need to be changed down the line.

However, function points are valuable in making early estimates, especially after the SRS has been completed.

Feature Point Size Estimates

A derivative of function points, feature points was developed to estimate/measure real-time systems software with high algorithmic complexity and generally fewer inputs/outputs.

A feature point or a feature set might have several function point attached to them depending upon the complexity and depth of software.

Feature set definition or feature point identification can be useful when historical data is available and average estimation is available or can judged for some commonly used feature points. However, in some cases the estimation of feature point and function point might be same.

Though features are easier to identify at early stages as compared to function point, the definitions might evolve over time during the course of the software and new details/function points might be identified afterwards.

Estimation Techniques

Once the software sizing is performed according to the unique requirements of any software project , efforts , resources and timeline constraints can be applied. After all estimates are in place a budget estimate can be done.

Estimation Considerations

Cost Drivers

Cost drivers are the variable parameters associated with every software project that affect the project cost/budget. There can be various cost drivers depending upon the specialized field of development however there are some basic drivers that need to be addressed no matter what type of software is developed.

Every good budget estimate must address the following variables as identified during scoping of the project:

- Number & Level of Resource (software developers, project manager, analysts, research, designers etc)
- Expected timeline (against the function points/use cases)
- Complexity level
- Technology/Platform constraints
- Domain constraints
- Customer centric constraints(deliveries, expectation, changing requirement etc)
- Type o f project(Time & Material, Fixed Price, phased etc)

The initial planning session should identify the major and minor cost drivers for the specific project to ensure that the estimates are safe enough. The cost drivers for engineering related products will be different from a large ecommerce solution as engineering applications are very algorithm specific, need more testing efforts, specific development environment experience and require more expert level of resources whereas a large scale ecommerce solution will only need to address large teams and changing requirements etc.

Risk Management

Every effective estimate should perform risk analysis to manage and mitigate any risks associated with a particular project. The level and volume of risk could differ from project to project but early identification leverages benefit to all involved.

Risks could be of any type such as new technology related, complex architecture, offshore or distributed team scenarios etc. These risks can raise the expenses such as hiring new resources, training , new hardware, testing & deployment resources thus will affect the actual estimates.

Contingency Planning

Contingency planning is a smart approach to risk mitigation and change management. Software projects are bound to changing and evolving requirements, special emphasis needs to be given to worst case scenarios where everything planned would go wrong. Contingency planning implies that the team has 2 main plans saying that if all goes well we will go with “Plan 1” but if “x” or ”y” or ”z” changes we will follow “Plan 2”.

However, contingency planning should not over budget the project that could bring shock to the client and expenses for the organization. Cushioning too many risks or worst cases into the estimates might raise the budget and timeline so much that the project appears to be a burden rather than a solution for the client.

Crisis Management

Crisis management is a very important consideration in case of offshore projects. The team needs to plan a very clear crisis management scenarios and outcomes to ensure smooth sailing of projects. This is main concern in time & material based projects and ongoing product development as long term commitments are involved and any cannibalization of targets or conflict of interest can result in serious crisis. Often political or similar event might also arise crisis that can only be solved as and when arises and is difficult to foresee earlier.

Though crisis management needs to be considered adding time or effort constraints or adding management costs for this task is not an intelligent approach.

Cost Benefit Analysis

Every estimation exercise needs to analysis if the project in question will really be profitable for the organization. The profit can not only be in term of monetary returns, though that’s the most trackable profit but also long term partnerships, team’s expertise, providing breakthrough in a new technology, market or platform are all valid profits for the organization. Any project that has a risk factor greater than the cost benefit factor is more prone to failures.

Rough Estimation

In some instances when a similar project has been done previously a rough estimation can be done based on the previous experience however this estimation would be required to be as close to reality as possible and should be refined as the planning matures.

Rough estimates are a good source of estimation when a prompt response is required, better estimates can be calculated once better understanding is developed. Such estimation will allow the planning team and the analysts to perform the entire planning and estimation tasks in detail as this exercise needs time and quick responses can affect the actual estimation process.

Outdated Estimations

In any real world project change is inevitable, change does not only imply change in requirements but it also implies change in project teams, resources, client’s target, partnership form/break. Whatever might be reason but these changed affect the estimates directly and thus with changes estimates will get outdated. Although its senseless to add margins in cost to accommodate this factor however, every project manager should address and make required adjustments to ensure that the change does not imbalance the entire project. Early notifications save a lot of efforts and misunderstandings. In some scenarios analysis of estimates to measure the deviation of actual versus expected can give managers more control over the project and this exercise can assist in developing measurement metrics for the organization.

Estimation On Project Type

Estimating Time & Material

Estimation in a time and material project is probably the easiest as the project itself is based on evolving and changing requirements therefore the initial estimates require addressing the right resources and technology constraints. However, the initial planning should consider how much efforts will be required by the associated resources so that the project does not overrun cost and time.

Estimating Fixed Price

Estimation of fixed price project is challenging and failure prone as requirements at an early stage are seldom completely known. Estimation through work breakdowns based on these initial requirements often result in incorrect numbers thus affects the actual time and budget allocated for the project.

There are some approaches to overcome these ambiguities in fixed price estimates to yield better numbers:

Estimate phase wise

Divide the requirements into phases and then estimate the efforts and budget for each phase. Phases simplify the estimation as well as are easy for the client to grasp when higher budgets are involved.

Breakdown and Prioritize

When breaking down the requirements into functional chucks always apply priority and complexity to each identified chunk. Estimating high priority and high complexity tasks first will allow to understand the overall nature of the project and will help in risk analysis.

Breakdown requirements in smallest chunks

When identifying use cases or feature points, try to break them into smallest possible chucks of functionality. This will allow easily estimation of the efforts, resources and time required to complete these tasks.

Estimate expected change of requirement

Through focused discussions with the management, project manager and senior developers a percentage of change of requirements can be identified by applying historic data of the organization. This percentage of change can be then considered by applying some cushion in the estimates to cover unknown requirements.

Analyze possible hurdles/bumps and speed breaker

Through similar exercise planning teams can identify expected hurdles or scenarios that might occur in future though are unknown at this stage. Issues that can arise during testing or deployment stages can affect time and effort estimated greatly and often left neglected. Identifying such issues might help making better estimates.

Understand the client's nature

The most important factors when estimating a project is the understanding of the client's business and nature. This can assist in indentifying the response delays, implicit expectations, possible changes and future scope of the project.

Estimating Offshore development

While estimating an offshore project, fixed price or time & material, there are certain parameters that need to be given special consideration to save over budgeted and overtime projects.

One of the reasons that offshore projects fail is that many times these projects are estimated and planned in a similar way as that of onshore projects. The time & space difference of such projects makes them difficult to handle and manage. If the differences are well understood and incorporated into efforts, chances of failures can be greatly reduced.

While estimating offshore projects following should be incorporated into estimates:

Communication hurdles and delays

Every offshore project will require more communication time as compared to similar onshore projects. Due to time differences the team will also come across response delays that will have to be considered while finalizing the timeline of the project. IF an offshore project is time&material based these delays can make the dedicated resource wait and stay idle until the client responds thus raising in-house cost for your organization.

Understanding of unspoken expectations due to cultural differences

Due to cultural differences, it is sometimes difficult to completely understand what the client said and what was perceived by the team. These unspoken requirements may cause delays and affect estimated effort.

Signoffs will take longer

Due to the nature of these projects it will take longer to take signoffs from the client and would involve more time of all involved in communicating with the client.

Higher costs of management

Management of offshore projects is very sensitive as compared to onshore projects. An explicit account of the management costs will be required.

It is very important to do a detailed **cost-benefit analysis** for such projects before finalizing the estimates as though the numbers might appear very attractive at first but the risks involved could create a complete disaster out of the project.

Offshore projects require more efforts in requirement specifications and scoping as long-distance discussion often arise ambiguities. When budgeting such projects special emphasis should be given to the time and delayed responses constraints in order to cover these expenses in timeline and cost.

One of the major problems with off shoring is that many buyers expect "lowest" cost and "quickest" responses while producing a solution that's I effective and of quality. The cost0benefot analysis should cover this constraint and ensure that the estimated timelines and costs are realistic and not too customer centric. It has been proven that projects that are estimated around the client's expectation only generally result in disasters, leaving no benefit for the vendor or the buyer.

Intelligent estimation involves considering the right parameters and every possible scenario and everything that can go wrong.

Estimating Breakthrough/Research based project

While considering a project that involves latest technologies, platforms or concepts that have not been utilized previously special focus is required for the learning curve of the current team members. A new development platform/technology might require extra efforts by the developers as well as the project manager for experimenting.

Such projects should be estimated similar to time&material based projects as no timeline can be defined upfront.

Estimating Agile Projects

This is probably the most talked about topic in software estimation. As agile development model emphasizes no requirement detailing or scope identification it is very difficult to estimate a project size without identifying the scope of the project.

It is understandable that the estimates for an Agile model the estimates will tend to change and the customer need to understand that as closely as possible since that's the nature of the process model.

As for the sizing a good approach could be to identify the building blocks of the system, roughly size each block, once a high priority block is selected for the agile delivery, detailed estimation can be performed alongside of the requirement analysis for that block. As the requirement are breakdown into user stories anyhow, the estimation depending upon these stories will yield even better estimates.

This way the customer can be informed of the size and efforts estimate for the block before development of that block is started and the actual essence of Agile model of short and executable iterations is saved.

Conclusion

There’s no standard or benchmark technique to generate the best estimates. There are different considerations and processes that can be combined with the best practices and historic facts of the organization to yield estimates that will be least likely to go wrong completely.

Software estimation is a difficult exercise and requires that the people involved in planning and estimation thoroughly understand the technical implications and limitation as well as the business scope of the project to be sized; only then reliable estimates can be generated.

The estimation drivers and influence factors can change from organization to organization depending upon the organization behavior, employed processes and the overall expertise and experience of that organization.