Estimation is one of the hardest things to do in software development whether using an agile methodology or not. There are a number of different ways to come up with the estimates, just as there are many different ways to prioritize the feature requests. The advantages of this approach are that it is very simple and can be done quickly. The disadvantages include a lack of precision and an inability to add up several stories into a meaningful measure.   
  
Ideal time is a useful metric because it is relatively easy for a developer to estimate, provided that the requirements/user stories are well defined and thorough. «Hours» are simply the number of hours that the development team estimates will be required to complete the user story. The disadvantage to using hours at this stage is that the team might be lacking information to make such a precise estimate. Because hours are so defined, there is a greater risk of executives and stakeholders placing inappropriate expectations on the team to deliver exactly in the estimated time frame.   
  
In Scrum, many teams use the concept of «story points,» which are an arbitrary measure that allow the teams to understand the size of the effort without the potentially binding expectations that come with hours. When estimating in story points, many teams use the Fibonacci sequence, a mathematical tool defined in the thirteenth century for rational approximations. The Fibonacci sequence is helpful for Agile estimating because as the points get higher, the degree of uncertainty is increasing. For the purposes of Agile estimating, the sequence has been modified to account for the needs of software development.   
  
Some teams have added a ½ for the user stories that are so small in effort they really do not need estimating. The difference between a 20, 40, or 100 is just a sense for the size of the epic. Secondly, some teams add 200, 500, and/or infinity . As that feature is better understood and broken down, it is eventually delivered in five months.   
  
Some members of the management team might now equate a 100-point story with a five-month delivery cycle, which is not true at all. One additional thing to keep in mind is that unlike hours, which have a fixed and commonly understood definition, story points are more abstract and negotiable. One team may assign 3 points to a story where that same story on another team would have been a 2 or a 5. As long as everyone on the team agrees to the size of a story point, then all of the estimating will be coherent.   
  
In some environments developers have to come out with fixed costs quotes for their customers. This is a widely used technique but unfair on your customer, you are basically adding a buffer on all requirements or stories regardless of complexity or detail that you have at the time. Instead Easybacklog.com use the 50/90 estimation technique which asks developers to estimate a story with 50% margin and one with a 10% margin . We type that in into Easybacklog.com for each story and it calculates a buffer for the backlog using the square root of the sums of the squared differences between the 50 and 90 estimates. 

Don’t worry Easybacklog.com calculates it all for you at the story, theme and backlog level. This gives you a better and fairer buffer on your estimates and costs as it will buffer where it needs it.   
  
**How does an Agile team actually execute on the estimation process?**  
  
In all instances, we are looking for participation from the developers and testers because we want to benefit from the wisdom of self-organizing teams. Outlined next are several strategies that teams have used to come to consensus on estimation. Having the development team participate in the activity of estimating can have distinct advantages over asking a single developer to size a feature. If the team is using story points one of the fun and effective ways to increase engagement is through a game called «planning poker».   
  
After the product owner and the team have discussed the user story in detail and everyone believes they have an adequate understanding, the Scrum master or coach will call for each team member to lay down the card representing the number of story points that he or she assigns to this user story. The Delphi method takes the same approach as planning poker but addresses the estimating in a more structured and formal way. A facilitator calls a meeting of the experts and a feature request is discussed. Each expert participant then completes an anonymous estimation form and submits it to the facilitator.   
  
Based on the new inputs, the expert team discusses the feature request again, clarifies any incorrect assumptions, gains a better understanding of the feature, and submits new estimation forms to the facilitator. The disadvantages are that it is time-consuming, and the experts are not required to reveal their biases to the group. The goal is to instill a degree of responsibility or ownership of the estimates to the developers. The XP process also provides a feedback loop for the actual time required, so developers can continue to improve their estimates over time.