

Agile Record

The Magazine for Agile Developers and Agile Testers

January 2011

issue 5

www.agilerecord.com

free digital version

made in Germany

ISSN 2191-1320

Early Estimation with Stakeholders

by Remi-Armand Collaris & Eef Dekker

Software development is hard, time consuming and expensive. No wonder the business wants to get it right the first time. Experience has shown that for this it is crucial to be Agile and embrace change. Embracing change gets cheaper when detailed specifications are postponed as much as possible, until right before a piece of functionality is built within an iteration or sprint. Only having high level requirements at the start of a project makes it harder to estimate the project. However, the business needs good early estimates to be able to build a solid business case for its software development initiatives. Agile methods acknowledge this: Scrum directs the team to maintain coarse-grained estimates for all items on the product backlog, XP states the following for the customer in its Bill of Rights “You have the right to an overall plan, to know what can be accomplished when and at what cost.”

Agile uses Planning Poker as its main estimation tool. Whereas Planning Poker works very well for estimating a Sprint or iteration, it is less useful for an early project estimation. For estimating Use Cases or Epics at an early stage (i.e. before they are detailed) we have used Use Case Points Analysis. In this article we share our experiences in applying this method in combination with an estimation game in which the team, users and other

stakeholders estimate the product backlog together. We presuppose knowledge of Scrum concepts like Planning Poker, Sprint Planning and User Stories (and Epics) or Use Cases.

The Estimation Game – Basic Version

Use Cases and User Stories are two forms of capturing requirements used in iterative and Agile contexts. Whereas Use Cases are built up from scenarios, User Stories are often organized into Epics. User Stories are mostly smaller than Use Cases and equal in size to scenarios. Epics are mostly equal in size or bigger than Use Cases. All of these can be used as units for prioritizing, estimating and detailing desired functionality.

To get early estimates on a project we invite stakeholder representatives and the members of the Scrum team to an estimation game. Have the User Stories and Epics or Use Cases ready, preferably on cards or stickies so that you can easily move them around on a board. Initially, these stories are on a stack, ordered by business value or some other form of prioritization so that the highest priority story is on top.

All participants stand in front of a board which looks like the one in figure 1.

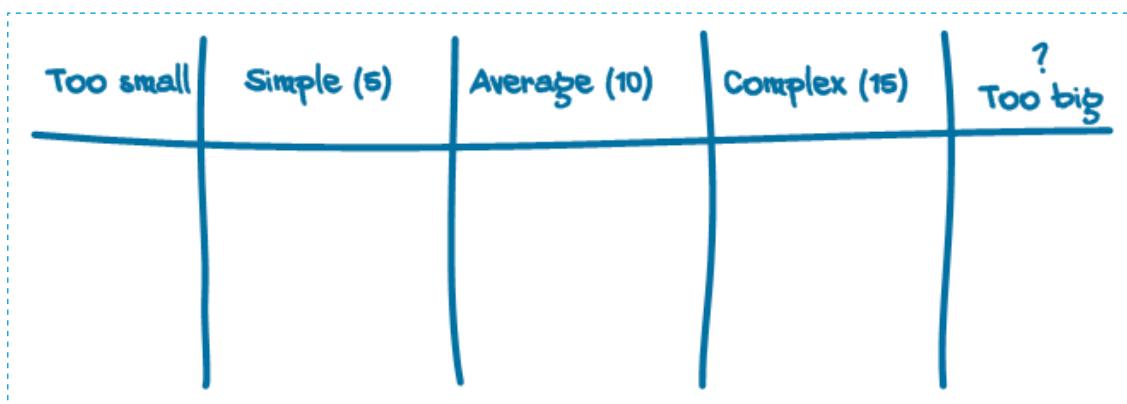


Figure 1: Estimation board – basic version

Each participant in turn does either of these things:

1. Estimate a new story from the stack by placing it in a column on the estimation board and explaining your estimation to the group.
2. Change an estimation by moving a story from one column to another and explaining to the group why you think this estimation is more appropriate.

Stop adding new cards to the estimation board some 20 minutes before the end of the timebox for this meeting (preferably two hours) has been reached, but continue changing estimations. The highest priority stories should be on the board by then and a new planning meeting can be scheduled to estimate the remaining stories on the stack. The game is over when all User Stories or Use Cases are on the board and/or none of the participants feels the need to relocate any more stories.

If a story is “ping-ponged” between two locations three times, remove it from the board and place it at the bottom of the stack. This story needs more investigation before it can be properly estimated. Stories in the “Too big” column should also be investigated more, be broken down into smaller stories or labeled as Epics that will be broken down later in the project.

This process makes estimation very concise and orderly. People have to await their turn to add or move a story, and they have to explain their actions to the group, furthering a better shared understanding of the story.

Expect extra Use Cases or User Stories to appear, and some to be split up or joined. This may be seen as choices the participants have during their turn but, in our experience, it is not necessary to state this as a rule. Participants will spontaneously do this when needed. New Use Cases or User Stories go to the stack and just follow the rules.

A very important advantage of doing an estimation game with the stakeholders is that they can share their view on the stories and hear why some stories are hard to build even if they seem simple from a user perspective. Making estimation a joint effort creates a shared ownership of the estimation. If stories turn out to be heavier later in the project as a result of added features not mentioned during the early estimation session, the impact is more easily accepted by the business.

The Estimation Game – Round Trips Version

In order to leverage experiences from past projects, it may be a good idea to involve not only the relative measures of simple, average and complex but to relate them to ‘round trips’ from the user to the system and back to the user. Suppose you have a user and a system with a screen on which the actor can interact with the system. A round trip then starts with a stimulus from the user when some action is input for the system. The system processes the input and returns the result to the actor. A new round trip starts when the user reacts to the result, which in turn is a new stimulus for the system.

The concept of a round trip initially came from the Use Case Points Analysis and was called a use case transaction, but you can use the concept just as well in estimating User Stories. Both User Stories and Use Cases may contain user – system interactions, and in both cases you can count the amount of round trips.

Not all Use Cases or User Stories contain round trips because they may not all be about user interaction. Such cases continue to be estimated relative to other stories on other criteria (as in the first version of the estimation game).

The estimation game is similar, but with one difference: a simple user interface based Use Case or User Story counts 1-3 round trips, an average one counts 4-7 round trips, and a complex one counts 8 or more. The estimation board now looks like figure 2.

Round trips:	1-3	4-7	8 ->	?
Size: Too small	Simple (5)	Average (10)	Complex (15)	Too big

Figure 2: Estimation board - round trip version

Participants have to think of the Use Case or User Story they talk about in terms of the number of round trips involved. This often gives an interesting perspective because the more user-friendly interaction you want, the more round trips you will need.

We have seen that people are fairly good at envisioning round trips and sharing their mental pictures of the system in term of round trips. When later on in the project it turns out that a Use Case or User Story is more complex than initially estimated, it is easy to go back to the original estimation and explain the diffe-



rence in terms of the amount of round trips involved. This makes it understandable to both stakeholders and team.

An example: In the early estimation session, a job vacancy search interface was regarded as simple; it was expected that the user would select search criteria from a couple of drop down menus and then submit his selection. Later on, however, it became obvious that the usability of the application would be enhanced if the system could already react to partial selections and update a counter showing the number of job vacancies found for the current search criteria. In other words, what was originally regarded to be one round trip turned out to be two. So, it first looked like:

- (1) The user selects search criteria and submits.

(2) The system searches for hits and shows relevant job vacancies.



Round trip 1

But then it was expanded as follows:

- (1) The user selects search criteria.
 - (2) The system updates a counter showing the number of hits.
 - (3) The user submits.
 - (4) The system searches for hits and shows relevant job vacancies.

You see the two round trips clearly here. Both stakeholders and team agreed that the initial estimation should therefore be adjusted.

From Round Trips To Hours

Once you have your user interface based Use Cases or User Stories estimated on the basis of the number of round trips and others estimated relative to these stories, it is possible to calculate effort in terms of hours. This can be done with the Use Case Points Analysis, which despite its name, can be used for User Stories just as well.

In the Use Case Points Analysis, the number and weight of the Use Cases or User Stories identified is the most important component in the calculation of the size of a system. You can balance this size by bringing in a consideration of the system's technical properties. The size of the system is the starting point for calculating the effort. Effort is balanced by considering the team's qualifications and other environmental influences¹.

This may sound a little complicated, but once you fill in the spreadsheet which accompanies the method, you will see that it is in fact quite straightforward². A simple User Story or Use Case amounts to 5 Use Case Points, an average to 10, and a complex one to 15. We have experienced that 20 hours of effort per Use Case Point is a good average.

The estimation obtained in this way is statistical information. On average, a story of the same weight will take the same effort but the actual time spent on each one may vary widely. Also in another sense an estimation in terms of Use Case Points is statistical: it does not work for a small system of, say, 5 stories. It needs at least 20 stories to be reliable. Nevertheless, our experience is that an estimation done in this way is closer to the actual effort spent on a project than an early expert estimation³.

¹ Detailed guidance can be found in our article Software cost estimation using use case points: Getting use case transactions straight (http://www.ibm.com/developerworks/rational/library/edge/09/mar09/collaris_dekker/index.html), published in the March 2009 issue of The Rational Edge.

2 Such a spreadsheet can be downloaded from several places, see for example www.scrumup.eu/downloads.

³ See on this Linda M. Laird, M. Carol Brennan, Software measurement and estimation: a practical approach, Wiley-Interscience 2006, p. 96.

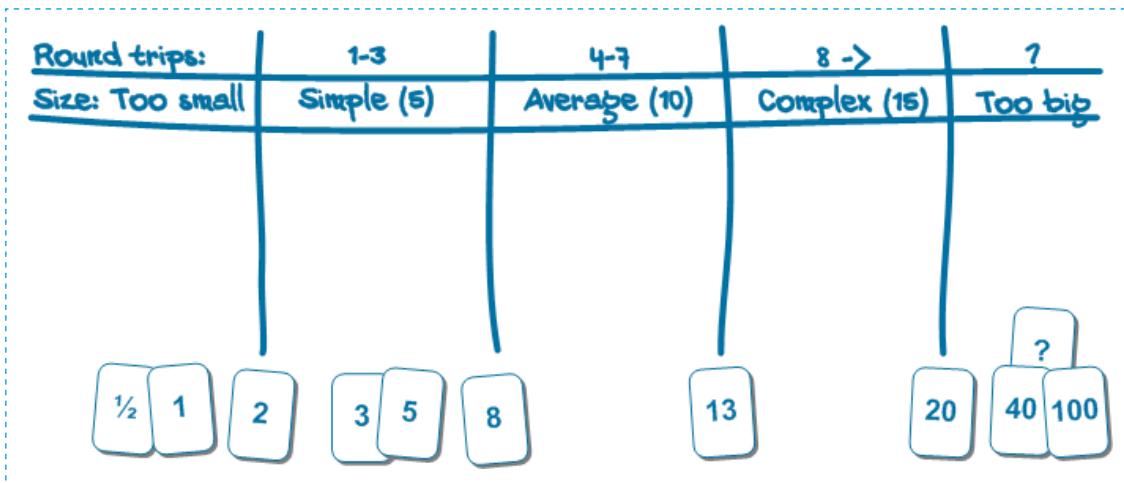


Figure 3: Early Estimation and Planning Poker

The Estimation Game And Planning Poker

The goal of the Early Estimation Game was to create an estimation for the whole of the project. Later on in the project, the team will use Planning Poker in order to get a more fine-grained team estimation in Story Points. This will be the team's basis for measuring the team's velocity and determining commitment for a sprint. Figure 3 shows how a cross-check can be made between the early estimation in Use Case Points (5, 10 or 15) based on round trips and the more fine-grained estimation in Story Points (0, ½ , 1, 2, 3, 5, 8...) resulting from Planning Poker. The Story Points can be taken to be a more fine-grained version of the Use Case Points. Now if it turns out in Planning Poker that a story estimate is wildly out of range, you know that your story has more functionality than was initially envisioned and you may need to adjust your release planning.

Conclusion

We have described a new approach for early estimation, which is straightforward and easy to apply. Just take turns and add an estimation or adjust one made earlier in the session. If you use the round trip version, it is possible to be more objective over different projects and over time, for you have the round trip as its basis. The advantage of having a measure of complexity that stakeholders can understand and having them participate in the estimation is that you share ownership of that estimation. We no longer have to struggle with stakeholders about size and necessity of changes, for they are formulated in terms they understand and estimated in a joint effort. ■

> About the authors



Remi-Armand Collaris

is a consultant at Ordina, based in The Netherlands. He has worked for a number of financial, insurance and semi-government institutions. In recent years, his focus shifted from project management to coaching organizations in adopting Agile using RUP and Scrum.

*An important part of his work at Ordina is contributing to the company's Agile RUP development case and giving presentations and workshops on RUP, Agile and project management. With co-author Eef Dekker, he wrote the Dutch book *RUP op Maat: Een praktische handleiding voor IT-projecten*, translated as *RUP Tailored: A Practical Guide to IT Projects*, second revised edition published in 2008 (see www.rupopmaat.nl). They are now working on a new book: *ScrumUP, Agile Software Development with Scrum and RUP* (see www.scrumup.eu).*



Eef Dekker

Eef Dekker is a consultant at Ordina, based in The Netherlands. He mainly coaches organizations in implementing RUP in an Agile way. Furthermore, he gives presentations and workshops on RUP, Use Case Modeling and software estimation with Use

*Case Points. With co-author Remi-Armand Collaris, he wrote the Dutch book *RUP op Maat, Een praktische handleiding voor IT-projecten*, translated as *RUP Tailored, A Practical Guide to IT Projects*, second revised edition published in 2008 (see www.rupopmaat.nl). They are now working on a new book: *ScrumUP, Agile Software Development with Scrum and RUP* (see www.scrumup.eu).*