Big Rock Estimation: Using Agile Techniques to Provide a Rough Software Schedule / Resource Estimate

This is the third article in the QSM Agile Round Table series. The QSM Agile Round Table was formed to discuss the role of estimation in agile environments. QSM customers shared their questions, challenges, and experiences on the relevance and benefits of scope-based estimation in an agile environment. The Round Table spent several meetings on the key topic of sizing an agile release. The discussion centered around two main questions:

1. *How can you determine the size of a release early in absence of a "big upfront requirements phase," and thus when the requirements are only known at a very high level and subject to refinement and change?*

2. *How can you determine size in a consistent way across multiple products, projects, and agile teams so that you have good historical data on which to base an estimate?*

This and the next article in the QSM Agile Round Table series are based on those discussions.

***Aaron Jeutter**, a participant in the Round Table from Rockwell Automation, presented the technique of "Big Rock Sizing." This technique is used at Rockwell Automation for early sizing and estimating based on high level requirements that will be refined using agile techniques as the work progresses.*

Introduction

Teams need to develop estimates for an overall schedule to support the initial funding and final funding milestones of projects. These teams within product development organizations are held accountable to these estimates.

Some groups at Rockwell Automation use a "Big Rock" estimation technique to derive a Rough Order of Magnitude (ROM) estimate for initial funding requests. The term "Big Rocks" came from a set of initial planning sessions completed for several projects. This method is also used to refine labor and cost estimates for the final funding milestone. The goal is not to estimate a mountain, or a whole bunch of pebbles… only the "Big Rocks". The estimates start with the same basic Agile premises:

1. The estimate of a group of people is usually better than that of a single person
2. The estimates won't be perfect. Some estimates will be high and others will be low. The estimate depends upon that balancing out in the end. The statistical property called the "Law of Large Numbers" often comes into play for most cases.
3. Force "choice points" of estimate values. If something is definitely "bigger" than a reference, it gets forced to the next size.
4. Align the activity to a common estimate schedule and do our best to ground the schedule with previous

project results.

5. Derive at least two different schedule estimates:
    a. Given a value of X number of developers, the possible delivery date is Y.
    b. Given a delivery date of A, you will need B developers to achieve that schedule.
6. Describe the Epics based upon expected scope, complexity and risk, not effort days.

Keep the sizing activities to a one or two day estimation session. The key participants involved with the sizing should be those who will be implementing the final product. Stakeholders outside of the team may participate and should be available to help clarify expected scope throughout the sizing process. Other non-technical participants (like project managers) may attend, but only as observers.

The basic agenda for Big Rock sizing activities consists of:

1. Introductions; Ground rules for the workshop; agenda walk through
2. Pre-work review
3. Sizing grounding
4. Estimation
5. Review and wrap-up

Pre-Work

The pre-work encompasses preparation of an outline and/or hierarchy of a preliminary architecture concept and clear definitions of the architectural elements. If an existing structure is not available, a small group of knowledgeable people should meet ahead of time to discuss and determine the initial structure. This structure does not have to be set in stone for the whole project. The key is to capture the information accurately and save it for reference in the future.

To keep the structure from getting too detailed, it is recommended to keep to three levels of Epic / Portfolio Item descriptions:

- First Level – Executive / High-Level description
- Second Level – Key Stakeholder, Manager-level description
- Third and deeper – Development Team / Engineer-level descriptions

It is highly recommended to keep the number of First Level Epics to about ten or less, but allow for more complexity at the Second or Third levels. Most of the time, Big Rock estimation only is limited to the second level of Epics. On occasion, building the Epics to the third level tends to happen with complex projects / products. Stories are the next level and should not be written until after the Epics are sized. The Stories should not be written in the Big Rock estimation meetings.

Sizing Grounding

The goal of performing the "Sizing Grounding" is to establish a common understanding of the values used when sizing the Epics. This can be one of the more difficult parts of the estimation activity. Teams almost always undersize the points, or are astonished when the points roll up to be so many, or the schedule runs so long. Many teams who do waterfall estimation grossly underestimate the schedule and then run late. The key is to focus on the *complexity and risk* in terms of Epic points and then normalize it back to our basis of guesstimated time from other projects. In many cases, a less complex time-consuming activity may end up having the same size as a highly complex, short duration activity.

At Rockwell Automation, most teams prefer to do T-Shirt sizing, especially for Big Rock estimation. Other teams use sizes of dogs (Chihuahua to St. Bernard) or other non-numerical ways of estimation. **Don't start assigning points to the sizes until after the T-shirt sizing is finished. Assigning points right away causes the team to only think about points and not comparative sizing of the Epics.**

The smallest T-shirt size should be considered to be more than the largest Story. Occasionally, a lot of smaller related features not big enough to be the smallest T-shirt can be collected into a "bulk" Epic. These are usually important items that can be completed in a very short amount of time or little effort.

The first step is to decide on what an Extra Small or Medium Epic is in terms of complexity, size, and risk. It's useful if some of the people in the room are experienced in Scrum and estimation and have a link back to actual data from a working scrum team. Since many teams like to have more granularity, it is recommended to add other T-Shirt sizes when appropriate.

The team should identify Epics or sub-Epics that everyone agrees is Medium-sized for the purposes of estimation. Select an Epic that is a likely candidate where the Epic is fairly well understood, discuss it, and see if the team can agree if the Epic is medium in size. If not, find another likely candidate Epic and try again until you find the Medium reference Epic. Once that is done, create brackets with definitions for XS and XL. The rest of the sizes can be defined as you start estimation.

An example commonly used at Rockwell Automation for Medium-sized Epics is the integration of Ethernet interfaces used by our products. Most teams reuse existing software or use toolkits to implement these interfaces. The scope and expected capabilities are similar across multiple products, making the Ethernet interfaces a perfect choice as a reference Epic.

Try to avoid defining the reference Epic in terms of effort hours, person days, etc. This is difficult to avoid, but the issue is that Developer A's effort hours to do this may be half of Developer B's.

Estimation

This activity is Rough Order of Magnitude sizing, not detailed estimation. The goal is to get through the Epics with a reasonable, but not perfect, estimate of Epic points with consideration of complexity and risk.

For estimation, the general steps are as follows

1. Look at the Epic. Is it small enough to size within 5 minutes? If so, skip to step 3.
2. If it's too big, or too complex, break it down into sub-Epics. Some of the sub-Epic work may already be captured via sub-requirements in the product requirements, the functional requirements or some other document.
3. Discuss the Epic, preferably in five minutes or less, to the point where the estimators understand the Epic.
4. The estimators state the size of the Epic. If there are a few very vocal people who are influencing other people's sizing, then you may want to use planning poker cards or another silent method. If necessary, the primary facilitator asks "Is this more or less complex than the Medium sized Epic?"
5. The high and low estimates are discussed. Uncertainty should be reflected in higher Epic sizing.
6. The estimators come to consensus on the final sizing.
7. The primary facilitator records the result, and the team moves on to the next Epic.

Review and Wrap Up

Once the team has finished sizing all of the Epics for estimation, the primary facilitator reviews the parking lot for anything that still needs to be discussed. The team is asked if there are any concerns or outstanding issues associated with the sizing. Everyone will want to know "what the answer is". Depending upon the tools used, the primary facilitator may or may not be able to give one.

The primarily facilitator takes the T-shirt sizing and converts it into Epic points. The facilitator should work with an expert estimator with experience in Scrum to do this. If possible, you want your Medium reference Epic to roughly match up with a known team's Medium Epic. The points for Medium can be extrapolated from this as a Rough Order of Magnitude (ROM) estimate.

When assigning points for Epic sizing, use sizes bigger than the largest Story that the teams plan to work on. For example: The team agrees that the largest Story is 20 points, so the smallest Epic would be 40 points. The modified Fibonacci series can be used to set the points for the remaining T-shirt sizes. The table below is based on this particular example:

T-Shirt	Epic Points
XS	**40**
S	**100**
M	**200**
L	**300**
XL	**500**
XXL	**800**

Reminder: Each team will have its own velocity. This velocity will be different than the assumptions made with the ROM calculations. The Epic points can (and should) be refined at a later date to match the real velocity of the team. Different teams will have different velocities, so resizing of estimates is essential for tracking and monitoring purposes.

Sizing Results Example

The table below represents results of a sizing activity for an application with PC and Web interfaces. The Parent Portfolio items are Communications, Business Logic, User Interface and Infrastructure. These are broken down into more specific, but still general work sets that can be sized. These lists are normally prepared as part of the pre-work prior to the sizing activity. This helps provide the structure for the discussions tied in with the sizing activity. For most projects, this will likely be a much longer list with at least one more level of complexity.

Portfolio Item (Parent)	Portfolio Item (Child)	T-Shirt Size (Pick List)
Communications	Ethernet Stack Integration	XS

Communications	REST Library Integration	S
Communications	Application Specific Features	L
Communications	Communications Testing	M
Business Logic	Database Services	M
Business Logic	Algorithm Updates	M
Business Logic	Business Logic Testing	S
User Interface	PC Interface Development	S
User Interface	Web Interface	S
User Interface	UI Testing	M
Infrastructure	DevOps Updates	S
Infrastructure	Build System Configuration	S
Infrastructure	Life Cycle Support	M

Once the sizing activity is completed, these can be assigned the Epic Point measures so an initial burn-up chart can be produced for review and discussion with stakeholders. Using the sizes indicated earlier in this document, this example works out to be 1940 Epic Points in size. Assuming the team is capable of completing 60 Points per Sprint and 3-week Sprints, this project could be completed in approximately 32 Sprints or 96 weeks.

Closing Thoughts

Teams and stakeholders to have discussions regarding the scope, complexity and risk for key functionality in the software product, without confusing that with schedule needs. When the schedule is considered first, the work estimate tends to magically compress to fit the desired schedule. Coming out of the first sizing session, the results will likely be unacceptable to key stakeholders. When this occurs, the project definition should be decomposed, redefined or refined to address the Epics that are considered too large. A subsequent sizing session should be performed after the project definition has been updated.

The Big Rock Sizing method balances estimation between the top-down and bottom-up methods used for sizing software efforts. Big Rock Sizing also provides the key stakeholders visibility into how the scope, complexity and risk exist in the proposed project. As the teams continue to use this method, Big Rock sizing can be adjusted to fit organizational needs. The results of this sizing can be used in conjunction with the SLIM Estimate tool using Function Units and the Agile estimation templates.

References

Steve McConnell, “Software Estimation: Demystifying the Black Art”, (Redmond, Microsoft Press, 2006), 115

http://www.scrumguides.org/scrum-guide.html