Estimation toolkit

Some useful techniques

James King

Version 1.0 – June 2010

Contents

Introduction

This toolkit consists of a number of approaches to estimating in both projects and “business as usual” work.

It is heavily biased toward agile projects and is designed to be used by people who have completed an agile project management course and/or been involved agile projects long enough to understand how they work.

The first section of the toolkit explains some general tools that can help the team when using story points to estimate. These consist of:

- Some tips for calculating velocity at the beginning of a project; and
- Some tools that support planning poker by improving the understanding the team have of the complexity of different stories.

However, we often need to estimate projects before we have even started the analysis on them. So I have included an approach to doing this. The approach does not produce a detailed estimate but it does provide an accurate enough guess to compare projects or to decide whether to start a project.

All of the techniques are ones that work well for me, but I will leave you to decide whether they are applicable or useful in your own environment.

Estimating according to agile theory

Agile theory generally recommends relative estimates. The team use planning poker to allocate story points to each story and compare these to the team’s velocity in order to plan the project.

Planning poker has been discussed in other places (including the Software Education “Agile Requirements - Stories” course) and we will not discuss it in detail here. However we will discuss:

1. Estimating velocity before the project starts; and
2. Tools to help the team better understand the stories being estimated.

Three ways to estimate velocity at the beginning of the project

The single most accurate way to estimate velocity on a project is to run several iterations and then see what the velocity was. This approach allows the team to find its rhythm and then use its experience in the real world to predict future iterations. However, those providing the funding for the project will often require an estimate before the project team can perform multiple iterations.

If the exact same team has been together for a similar project using the same tools and technologies then they might provide an estimate based on the previous project. Again, however this is often not practical.

So there are three ways that commonly work well and I have described them in the next three sections.

Using traditional methods after the completion of planning poker

One of the great advantages of planning poker is that it flushes out the assumptions and constraints involved in the solution.

This information provides good input for the use of the senior team members who traditionally do the estimating, so rather than trying to convert story points to time and money, the estimators simply use that information as an input into their calculation of the duration of the project (for example the team could use the WAG generator described later in this document.

This may seem like duplicated effort, but it might provide the team with confidence in their first couple of agile projects.

Once the team has an estimate for the whole project then they can decide to receive the project, gain funding and then begin the project with the same confidence that they usually have when starting. Then the team will be able to run several iterations and adjust their estimates as they gain more information. Once this has been done then the team can determine and share their velocity.

Working backwards from the iteration length

The second approach to choosing a velocity is to determine the iteration length and team size, and then decide how many stories the team could complete in an iteration. This is done as follows:

1. Decide on the iteration length, as well as the expected team size and composition.
2. Bring the team together and select some sample stories (these need not be the stories that will be done in the early iterations).
3. Allocate stories to the first iteration one at a time and let the team determine how many of them they think they could get done in the time available:
    - This can be done by simply moving stories in and out of the iteration until the team are happy they have “about one iteration of work” allocated; or
    - Stories can be broken into tasks and those tasks can be added to the iteration. This involves estimating how many hours each task will take and simply adding them together until we consume the total team hours available in the iteration.
4. Repeat the process for a couple more iterations if you feel this will increase the accuracy of the estimate.
5. Once you have worked out how many stories fit into an iteration they the total of the story points for each story will be the estimated velocity.

Working upwards from the time taken for one or more stories

The third way to estimate the velocity is to work out how long it takes to deliver one story point and then use this number to work out how many story points can be completed in an iteration. This is done as follows:

1. Choose a typical story.
2. Determine all the tasks needed to complete the story and covert these into expected person hours.
    - Add up all the hours needed by each person in the team;
    - Use ideal time (effort) rather than elapsed time;

- For example, if a story would take 1 hour of a BA's time, 2 hours of a developer's time and 1 hour of a technical writer's time then this would translate into 4 person hours.

3. Divide the number of points in the story by the number of person hours to find out how many points can be completed in one hour.
4. Multiply the figure you have calculated in step three by the number of hours in the iteration to determine how many points could (in an ideal world) be completed in an iteration.
5. Divide the figure you have calculated in step four by two in order to convert from ideal time to elapsed time. This is the velocity that is predicted by the one story.
6. Repeat the process for more stories and average the results. Estimating additional stories will add to the accuracy of the estimate, but generally after 5 stories you will have found a good estimate of the velocity if you chose a range of different stories and realistically broke them into tasks.

Four tools to assist with estimating stories

Things that matter chart

The "Things That Matter" matrix ("TTM") is a simple tool for exploring the complexity of a story. a TTM consists of a list of the technologies that are needed to deliver a series of stories. The technologies involved are listed on the left of the matrix and stories are listed on the top, as shown in the table below:

		Stories				
		1	2	3	4	5
Technologies involved	HTML	X				
	Silverlight	X		X	X	X
	SQL	X			X	X
	XML		X	X	X	X
	Firewall integration		X			X
	Java				X	X
	CMS			X	X	

The aim of the TTM is to allow the group to understand the complexity of story by seeing which stories involve which types of technology. The theory here is that a story crossing many technologies will be more complex and thus larger than stories only involving one or two types of technology.

Adding a little complexity

The effort involved in some stories is dependent on more than just the technologies involved. So rather than just listing technologies we may choose to list any significant "things that matter" such as:

- Complex testing (scalability, usability, security etc);
- User research; or
- Legal sign-off.

Rather than simply listing whether a technology or process is involved, we may list whether the impact is simple or complex. So, for example we could replace the "x" in the boxes with the following:

- H = High complexity or effort;
- M = Medium complexity or effort; and
- L = Low complexity or effort.

This would give us a slightly more complex TTM as follows:

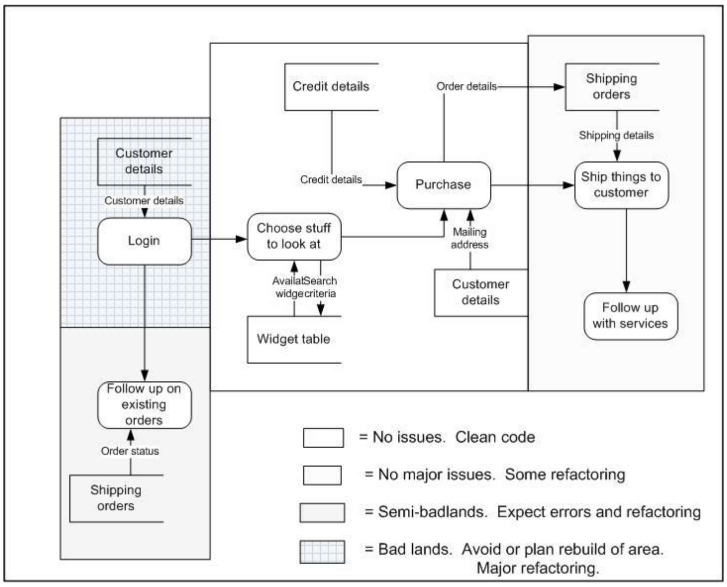
		Stories				
		1	2	3	4	5
Technologies involved	HTML	L	H		H	L
	Silverlight		H	M	M	
	XML	M		M	M	L
	Wireframes	M		M		
	Java		H			
	CMS		H			
	User research		H	M		
	Integration testing		M		M	
	Regression testing		M		H	
	SEO changes					
	Changes to contract				H	
	3rd party involvement		H		H	
	Training for sales team		H	M		

System heat map

A heat map shows where to expect trouble in a system and where to expect smooth sailing.

To create a heat map, simply take any model of the system and shade or colour different areas of the model to indicate the ease or complexity of working on different each area. Things that can be taken into account include whether there are automated tests in place, areas of spaghetti code, areas that involve complex integration code, areas the developers know well or anything else the team feels is relevant to the complexity (and therefore estimated effort) involved in working on different areas of the system.

The team adjust estimates involving different areas of the system in line with the "heat" of the particular area.



Lo-Fi prototypes

A Lo-fi prototype is simply a picture or a simple model of the system. These can be used to create new stories or to help the team provide estimates.

A measure of uncertainty

It is easier to estimate work that we are familiar with than things we have no experience with.

We can formalise this by coming up with an uncertainty rating (Jim Highsmith calls this the “exploration factor” in his book “Agile Project Management”. For example people might classify stories as

- Reusing existing solutions;
- Familiar work; and
- New and untried solutions.

Estimating with insufficient information

The WAG generator

The WAG generator (also known as “wide angle guess” or “wild a***d guess” tool) is not in common practice, in other words it works well for me but there is no research or alternative sources to confirm it works for others. Therefore you should consider the idea on its merits and use it if you see value in it.

The purpose of the WAG generator is to provide the estimator with a rough and ready estimate that may be accurate enough to make decisions in the absence of time consuming analysis. I have often found this to be necessary at the beginning of projects or working out which enhancements deserve further investigation and which should be abandoned straight away.

The estimate is provided in either days or dollars and is done by selecting a range from a table rather than allocating a specific value to the task.

Creating the generator

The WAG generator is simply a table containing letters that represent a range of values.

- “A” is generally taken to represent a number big enough for the customer to refuse to commit to the work without a more detailed estimate, and thus the funding of some further analysis;
- “B” is allocated to items that are about half as much as “A”;
- “C”, “D” and the following letters are then allocated to items that are successively smaller; and
- Once items reach a trivial level then no more letters are added.

Using the generator

Once the WAG table is created then new items are estimated using the table rather than being analysed properly.

Based on the WAG estimate, the customer can:

- Decide not to proceed;
- Commit to proceed;
- Prioritise between multiple items without needing further analysis; or
- Commit to further analysis if it appears justified.

This allows the estimator and the customer to quickly shift through a large number of items without allocating substantial effort to any of them.

Enhancement WAG

The first version of the WAG generator is used to provide a quick and dirty estimate for small requests made to a business as usual team.

Rather than sitting down and analysing requests, the estimator allocates the item to wide range of dates (or costs). This estimate may provide enough accuracy for an initial prioritisation session, after which the team may prepare a more accurate estimate if needed.

Rating	Estimated time
A	Over 6 months
B	3 – 6 months
C	4 – 12 weeks

Rating	Estimated time
D	2 – 4 weeks
E	2 – 10 days
F	1 – 2 days

Rating	Estimated time
G	1 – 8 hours
H	¼ - 1 hour
I	Less than ¼ hour

Note that instead of time, it is also possible to use the approach with financial estimates. For example the by allocating the letter A to an estimate of over $1m, B to an estimate of between $500k to $1m and so forth.

Project WAG

The project WAG is similar to the enhancement WAG except that it is a little more complex.

The complexity comes from the following additions to the standard WAG:

- The estimate consists of a series of smaller estimates;
- The estimate includes a range similar to the traditional “best case, worst case and most likely case” estimates. However it starts with the most pessimistic estimate and then works backward to find a rough guess; and
- The estimate explicitly references key risks.

Based on the WAG, the project manager and the customer might decide to:

- Proceed with the project;
- Reject the project; or
- Analyse and estimate only some components in more detail as these represent the biggest risk, the biggest cost or the biggest range in the estimate.

The best way to explain the use of the project WAG is through an example.

WAG example

A project manager needs to provide estimates for the following project:

Objective: To deliver a data tracking system that automates reporting for sales managers.

In scope	**Out of scope**
System changes to both front end and backend systems	Rollout to other Australian states
Rollout to offices in NSW, Vic and South Australia	Process change required as part of the project
Delivery of training material to users	

Step 1 –Gather the available data

The PM must first break the project down into meaningful components. For example:

- High level tasks, milestones or deliverables;
- High level features or stories; or
- The project scope.

The project manager decides that in order to provide an estimate he will break the project into the following relevant components based on the scope and some critical tasks:

- Analysis of management reporting needs;
- Delivery of training materials;
- Front end system changes;
- Backend system changes;
- Rollout to offices in NSW, Vic and South Australia; and
- Building data interfaces to existing systems

Step 2 – Select a rating scale

The project manager creates a WAG generator. In this case our project manager has chosen to estimate in time based on the following categories:

Rating	Estimated time
A	Over 6 months
B	3 – 6 months
C	4 – 12 weeks

Rating	Estimated time
D	2 – 4 weeks
E	2 – 10 days
F	1 – 2 days

Rating	Estimated time
G	1 – 8 hours
H	¼ - 1 hour
I	Less than ¼ hour

Step 3 – Add the components to the table and complete the 99.5% confident column

The next step is to add the components that we are going to estimate. The table below shows the components added to the table.

Now we create the first estimates. Project managers tend to claim that they cannot provide an estimate without doing the analysis stage of the project. But this is not strictly true. What they mean is that they cannot provide an estimate that the customer would like to hear.

It may sound absurd but if we allow the project manager to be as pessimistic as he or she wants to be then it is always possible to provide an estimate that the project manager could be 100% confident of meeting. For example, I would be almost 100% (say 99.9999999%) confident of:

- Launching a new company intranet in less than 10 years; or
- Creating a new blog in Wordpress in less than 1 day.

So the issue is not really that we are unsure how to estimate, it is that we need to either add ridiculous contingency to our estimates or admit that we are less than 100% confident in them.

This is the basis of our WAG. We will start by providing the smallest number that we are pretty sure (99.5%) we can meet. This means that there is a chance we would not deliver in the time frame but we are almost certain we could do it.

At this point you might choose to add risks to the estimates, but this is optional.

First cut of the WAG generator

Item	99.5%	60%	Guess	Key risks
Analysis of management needs.	A			Need to lock down the scope before committing.
Delivery of training materials.	B			Training will depend on process changes.
Front end system changes.	C			
Back end system changes.	A			We have not looked at this component – so we have no idea what is involved.
Rollout to offices in NSW, Vic and South Australia.	B			
Building data interfaces.	C			

Step 4 – Break down the items that are too big to provide a meaningful estimate

On the first pass of the estimates, the project manager identified several items as being over 6 months. This is too large a timeframe for the purpose of estimating.

So we break these items down to clarify dependencies, understand the task better or otherwise bring the item down to a point where we can provide some sort of estimate.

Second cut of the WAG generator

Item	99.5%	60%	Guess	Key risks
Workshop to scope management reporting needs.	D			
Follow up analysis on specific management reporting needs.	C			Unforseen reporting requirements lead to scope change - impacting budget.
Confirmation of reporting needs with CFO and Head of Sales.	E			
Confirm process changes with the Sales Operations Team.	E			Lack of confirmation means training material cannot be prepared – impacting schedule.
Delivery of training materials.	D			
Front end system changes.	C			
Data warehouse integration	B			Data warehouse integration not practical – impacting budget and system design.
Mainframe system changes	C			
Rollout to offices in NSW, Vic and South Australia.	B			
Building data interfaces	C			

Step 5 – Add the 60% confidence estimate

We now have a pessimistic but not impossible estimate. But we also want to know how long the project is likely to take, rather than just how bad things could be.

But rather than looking at what range we would be 80% or 90% confident of meeting, we pick one that we are only 60% confident of meeting. I am not sure if I can back this up scientifically, but I have think that:

- People tend to enter the same range for 80% confidence as they did for 100% confidence. But we actually hope to do better than our worst guess;
- If we aim for 50% confidence then people become concerned that we are not really committing to anything; and
- Picking 60% gives the ability to estimate without fear, while also committing to a number we think is more likely than not to be achievable.

You can choose a different confidence level if you believe it is useful.

In addition at this stage we add more detail in the key risks and dependencies that the 60% confidence guess depends on. These should particularly be added to any items where there is a large difference between our 60% and 99.5% confidence estimates.

In our example the project manager has added the 60% confidence estimates in the table and also added risks. He has also added one more item (pilot rollout) to mitigate the risk he identifies in rolling out the solution to multiple states.

Third cut of the WAG generator

Item	99.5%	60%	Guess	Key risks
Workshop to scope management reporting needs.	D	E		
Follow up analysis on specific management reporting needs.	C	E		Unforseen reporting requirements lead to scope change - impacting budget.
Confirmation of reporting needs with CFO and Head of Sales.	E	F		
Confirm process changes with the Sales Operations Team.	E	E		Lack of confirmation means training material cannot be prepared – impacting schedule.
Delivery of training materials.	D	D		
Front end system changes.	C	C		
Data warehouse integration	B	C		Data warehouse integration not practical – impacting budget and system design.
Mainframe system changes	C	C		Lack of resources delays project.
Pilot rollout in NSW	E	E		
Rollout to offices in NSW, Vic and South Australia.	B	C		Miscommunication or misalignment around training, system change and process changes leads to poor experience.
Building data interfaces to existing systems.	C	C		
Overall estimate	A			

Step 6 - Complete the WAG generator

The final step is to complete the generator by adding information to the remaining fields in the table:

- Add a row for each of the following and include them in the estimate:
  - Business case, project approval and project establishment;
  - Project management effort of 10% if you want to run streams concurrently; and
  - Project deployment, handover, operationalisation and closure
- Add up the worst case values for the 99.5% column and then adjust for the fact that many tasks can be completed in parallel. Take a guess as to which letter best represents the total project duration/cost and add this to the bottom of the table;
- For each item, guess how long the item will take to deliver in days or weeks (or how much it will cost in dollars). This should be a figure from the range shown in the 60% confident column;
- Add the numbers in the guess column and adjust for items that can occur in parallel or a little contingency and enter the number at the bottom of the table in the square provided; and
- Provide an estimate along the following lines:
  - An optimistic guess is that it will take/cost (the total from the guess column) but it could be as much as (the total for the 99.5% column);
  - Subject to the following risks and assumptions (show the risks from the risk column); and
  - If you want to know the critical areas that are impacting the estimate then they are (list the items that have the biggest impact or carry the largest cost).

Third cut of the WAG generator

Item	99.5%	60%	Guess	Key risks
Workshop to scope management reporting needs.	D	E	4 days	
Follow up analysis on specific management reporting needs.	C	E	5 days	Unforseen reporting requirements lead to scope change - impacting budget.
Confirmation of reporting needs with CFO and Head of Sales.	E	F	2 days	
Confirm process changes with the Sales Operations Team.	E	E	6 days	Lack of confirmation means training material cannot be prepared – impacting schedule.
Delivery of training materials.	D	D	15 days	
Front end system changes.	C	C	30 days	
Data warehouse integration	B	C	45 days	Data warehouse integration not practical – impacting budget and system design.
Mainframe system changes	C	C	30 days	Lack of resources delays project.
Pilot rollout in NSW	E	E	5 days	
Rollout to offices in main states	B	C	40 days	Miscommunication or misalignment around training, system change and process changes leads to poor experience.
Building data interfaces to existing systems.	C	C	35 days	
Project management			25 days	
Business case and setup	C	D	10 days	
Handover and closure	C	C	25 days	
Overall estimate	A		277 days	Recommend kill the project.

How accurate is the project WAG?

This estimate is not very precise. But it is often accurate enough to compare projects, highlight where further analysis will be the most beneficial, understand the riskiness of the project and make decisions about projects.