

I have started many personal projects over the years. I work on them on the side of my day job, usually at odd hours of the night or on weekends. I work on some projects for a few days or weeks and then I lose interest or get bored. For a few projects I have managed to take them from an idea to completion. For an even smaller subset I ended up with a product that some people wanted to use and even pay money for.

I believe that of all the projects I started, the projects that I completed were those for which at some point I got serious and started to maintain a schedule and manage my time.

For me, having to constantly make decisions on what to do next as I finish tasks is what takes the fun out of the project, so I am more likely to drop it. I have found that if I do as much of that planning and decision making work as I can up front, then it becomes easier to complete a project, it feels as if it is actually somebody else that is handing me pieces of work that I just need to complete, so I just get to do the fun stuff. When I finish a task, I just grab the next one in the list, without giving it much thought. Funny how the mind works. Well, at least mine.

There's another aspect to consider. For many types of projects there isn't really a clear end. This is almost always true in software projects. When a software project does become successful, even if you managed to stay away from planning and schedules, there will be real people out there using it, and there will be feature requests and an upcoming version 2.0. If a project reaches this stage, users will invariably start asking what at first sight seems to be an innocent and simple question, but one that small teams or independent developers will have a hard time to answer accurately: "When will *<insert a favorite feature here>* be available?".

I hate that question. After all, I work on personal projects because they are fun. The fact that a personal project turned out great and became a real product that generates me some additional income is irrelevant, I have to answer that question every day at work and now I have these strange people that call themselves users of my product after me as well?

Here too, I have found that using simple project management techniques help. In the past I have provided eyeballed answers to this question, or simply lied about it. Having a schedule in place does help come up with an answer that is much closer to the truth.

So what is this Project Management business, you may ask?

There isn't really much to it. You can manage your project and your time with a simple Excel or Google Docs spreadsheet.

Project Management For Dummies

If you wanted to implement the simplest approach, you can manage your project by just making a list of the tasks you have to do to complete it, and keeping track of which tasks are done and which tasks are still to do. When you finish a task you just remove it from the list and then look for another task to continue. If at any time you realize you missed a task, you immediately go back to your list and add it, before you forget. Pen and paper is all you need for this.

I started tracking my projects with the pen and paper approach described above, but over time I have been evolving and refining my method. In the present I use a system that borrows some ideas from the Agile software methodologies. Even though the ideas come from the software world, they can really be applied to any kind of project.

My current method is still extremely simple, and it can be implemented in a spreadsheet. Before you continue, please [ctrl-click here](#) to open a Google Docs sample spreadsheet with my example schedule. The following sections describe the different parts of this spreadsheet.

The Backlog

Backlog			
Tasks	Release	Estimate (Hours)	Iteration
Web page layout			
Base HTML Template	Beta	4	1
CSS	Beta	4	1
Database			
Design user table	Beta	8	1
Define user ORM class	Beta	4	
Define forum tables	1.0	16	
Define forum ORM classes	1.0	6	
User profiles			
Login/logout subsystem	Beta	16	
User profile page	1.0	8	
Edit user profile	1.0	16	
User stats page	2.0	8	

The *backlog* contains the list of tasks that you need to accomplish to complete your project, plus some data about them.

In the example spreadsheet, make sure you click on the *Backlog* tab at the bottom left if that is not the current tab. For this example I have defined three task groups with ten tasks distributed among them.

The columns I have in my Backlog table are:

Tasks

This column is straightforward, it just contains a short description for each task.

Release

The Release column contains a release tag for each task. For me this is very useful, as many times I come up with ideas for my project that are not really something I want to get done right away but more of a future enhancement thing. Having the planning for two or three releases in the backlog allows me to keep track of these ideas for the future.

You can use any text to represent a release. In my example I'm using "Beta", "1.0" and "2.0", but you can use any labels you like. As you add tasks to your backlog, just pick the release that makes more sense. The release choices that you make aren't set in stone; you can move tasks between releases as often as you want.

Estimate (hours)

The task estimate is a very important piece of information that is used to answer the "When am I going to be done?" type of questions.

You should make your best guess as to how long each task will take you. It is hard to estimate accurately at the beginning, but the more you do it the better you get at it.

I like to write my estimates in work hours, but you can really use any time units you like. It is important that the estimates are in linear time, without considering things such as how many hours a day you work, or your holiday schedule.

If a task is so complex that you end up with a really large estimate, or maybe you can't even come up with a number, then that is a good indication that this task needs to be broken down into smaller pieces.

I have found that I make more accurate estimates when I deal with short tasks. In practice, any tasks that I think will take me more than a week to complete are tasks that I have to break down into smaller subtasks. For example, if you devote about 16 hours per week to your project, then you should break down your tasks so that all of them have an estimated work time of 16 hours or less.

Iteration

When you add a new task to the backlog just leave the iteration column empty. In the beginning your backlog will not have anything at all on this column. The next section shows how this column is used.

The Iterations

Iteration #1	Start Date:	6/8/2012
Tasks	Release	Estimate (Hours)
Web page layout		
Base HTML Template	Beta	4
CSS	Beta	4
Database		
Design user table	Beta	8

The development of the project is done in fixed chunks of time called *iterations*. To begin an iteration, you just pick the list of tasks that you think you can complete during the iteration period, and then just get those done. It's really that simple.

Before you can start assigning tasks to iterations you have to decide how long your iterations will be. Again, do not consider weekends and holidays for this, just go with calendar days. I have found that for me using 7 or 14 day iterations is what works best. Four week iterations are also pretty common, but I find it more difficult to make good estimations with such a long period of time. In the example spreadsheet I went with 7 day iterations.

After you decide on an iteration length you have to estimate how many hours you will work during the iteration. The hours you will be spending on an iteration can vary from one iteration to another, so this is something that you have to decide when you are ready to begin one. For example, if during an iteration period you plan to be away you have to account for that. To continue with our example spreadsheet, let's say that we will be working about 16 hours for the first iteration. How you distribute those 16 hours among the 7 days is not important.

To assign a task to an iteration you start by entering the iteration number (starting from 1) in the iteration column of the backlog. Picking tasks is an easy process, you should just select the tasks that you want to work on next, making sure to not go over 16 hours, at least not by much. In the spreadsheet I have assigned three tasks to my first iteration, totaling exactly 16 hours.

After the tasks that will be part of the iteration have been identified, we want to copy them to a new table that is specific to the iteration. I do this by creating another spreadsheet and copy/pasting the selected tasks to it. In the example you can click on the "Iteration #1" tab at the bottom left to see the how the iteration table looks for the first iteration.

Note that I have only copied the first three columns of the backlog, clearly it makes no sense to copy the Iteration column since this new table is specific to an iteration.

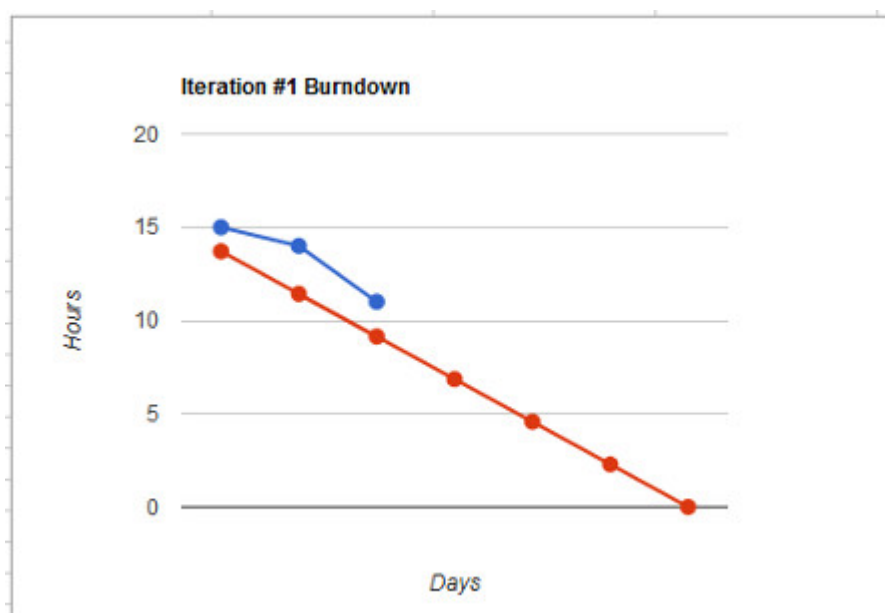
The bottom of the table contains the totals. This is all driven by formulas so you can just let the spreadsheet calculate this stuff for you. The bottom row, labeled *Ideal Burndown* is also all calculated with formulas so you can just pretty much ignore it. We will be using this data for the burndown charts.

So as days go by you just enter your numbers. If everything goes well by the time you reach the last day of the iteration you will be writing zero on all tasks, indicating that you completed all the scheduled work. So just congratulate yourself for meeting the deadline and then begin a new cycle with the next iteration.

What happens if you reach the end of the iteration and you haven't finished some of the tasks? Nothing, really. Your schedule had just been delayed a little. In such a case you take the unfinished work and copy it to the next iteration, along with new tasks, until you fill your hours per iteration again. An important aspect of the scheduling process is that no matter what happens iterations have always a fixed length. Any work that you couldn't finish on the current iteration, you have to move to the next.

If iteration after iteration you find that you always end up with some unfinished work then this could be an indication that the hours per iteration that you think you can give to your project is too high, or else your estimations are too low. You may consider assigning less hours of work to future iterations so that you can hit the deadlines more accurately. Conversely, if you find that many times you reach zero sooner than the last day and you end up with nothing to do for a while you may want to load up your iterations with more work. It's a process that improves over time, after a few iterations you will find the right balance.

Eye Candy - Iteration Burndown Charts!



The burndown chart gives you a quick idea of where you are during an iteration. While this isn't really a necessary part of the process, the charts are very helpful in visualizing your progress.

The chart's X axis represents iteration days, so in our example it has seven marks, one for each day of the iteration. The Y axis represents estimated hours of work that are left in the iteration.

The red line represents an ideal burndown rate for the iteration. This simply takes the total estimated hours for the tasks in the iteration and spreads them evenly across the days of the iteration, so in our example each day the amount of work left is reduced by 1/7th of the total, arriving at the last day with nothing left to do.

The blue line is plotted from the burndown numbers that you enter day by day. By comparing the blue line to the red line you can easily tell if you are behind or ahead in your schedule. When the blue line is above the red line you are behind, and when the blue line is below the red line you are ahead. The goal of the game is to reach zero by the end of the iteration. Pretty simple.

Once you accumulate a few iterations you can look at the burndown charts to identify any adjustments to the hours per iteration that you can consume in each cycle. You may start to realize that while the estimates were done in work hours, at some point we really don't care anymore that these numbers represent hours, they are just a measure of the task *complexity*, and you just need to find how much complexity units you can consume on each iteration. In the Agile world it is actually common to estimate tasks in *points*, which are an abstract measure of complexity. I find it easier to use hours, even if my estimates end up being way off.

The Release Schedule

Release Schedule			
Release	Hours Left	Hours per Iteration	Iterations Left
Beta	20	16	1.25
1.0	46	16	2.875
2.0	8	16	0.5

For this final part we are going to go back to the Backlog spreadsheet. To the right of the backlog we have the release schedule table.

This table estimates how much work is left to do after the current iteration. The table sums up the hours from the backlog for any tasks that are not assigned to an iteration, grouping the results by release. After you enter your estimated hours per iteration in this table you can get an idea of how many more iterations you need to complete each release.

For me this is the most valuable piece of information, since it helps me plan each release. For example, if I wanted to have a shorter Beta release I can either find tasks to move out of the beta release and into a future release, or maybe decide to give more time to the project, so that I can do more on each iteration. So I just start playing with the release tags and the hours per release numbers until I get each release configured the way I want.

Conclusion

I hope this was a useful introduction to project management. If you have any comments or questions about the topics covered, or if you have your own project management method that you would like to share, please do so in the comments below.