

Managing Risk on Agile Projects with the Risk Burndown Chart

by Mike Cohn • 68 Comments

Risk management is a central part of traditional project management and is included as one of the knowledge areas in the Project Management Institute's (PMI) body of knowledge. In many of my classes, participants ask how Scrum and agile address risk management. Some are concerned that agile or Scrum ignore risk management completely. Let's see why this is the case. First, a great deal of explicit risk management becomes unnecessary when a project uses an agile approach. The short iterations, single-minded focus on working software, heavy emphasis on automated tests, and frequent customer deliveries help teams avoid the biggest risk most projects face—that of eventually delivering nothing. So, it is not surprising that many agile projects forgo any form of explicit risk management. To put it in risk management terms, for these projects the likely savings from explicitly managing risks is outweighed by the cost of explicitly managing risk.

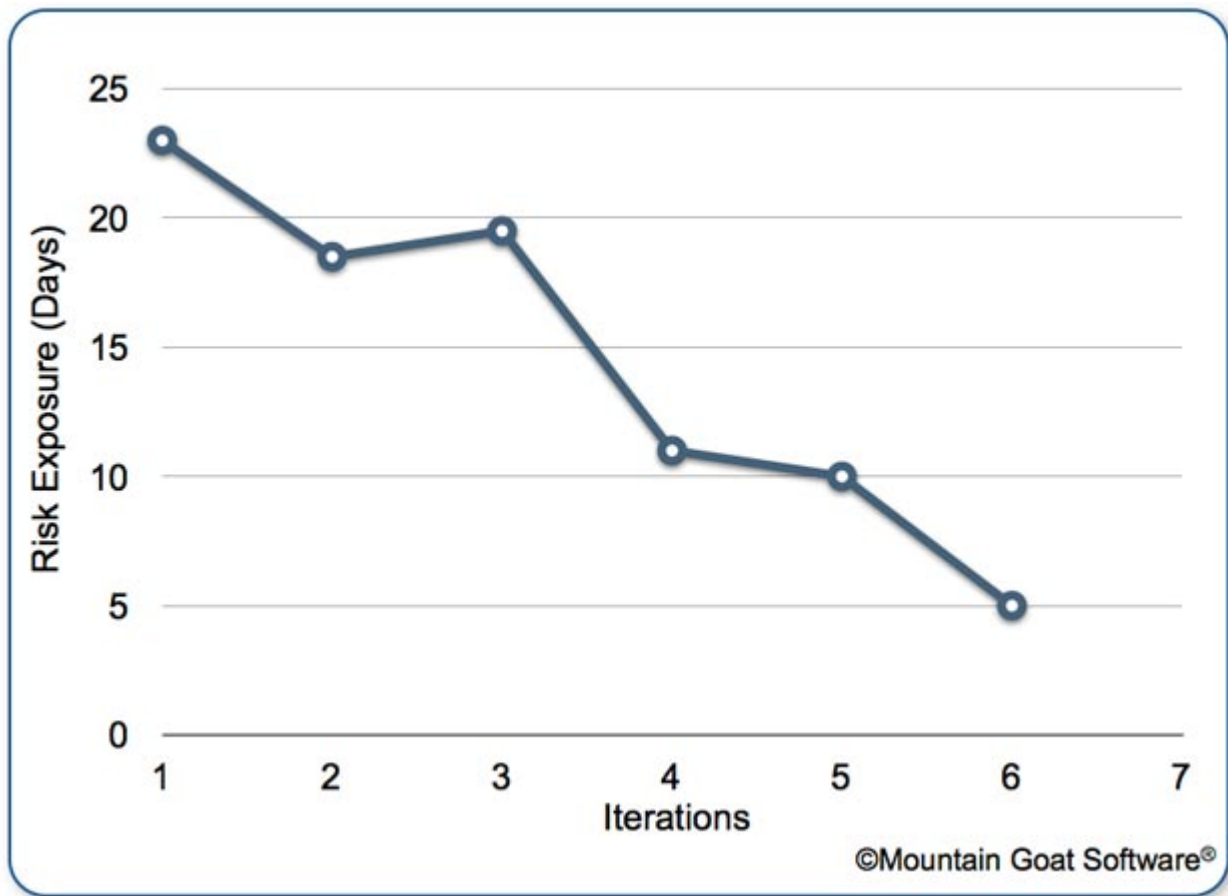
So while some short and inherently low-risk projects can be safely undertaken without any formal risk management, other projects are likely to benefit from explicitly managing risk. To see how we can do this, I want to describe a technique originally introduced by John Brothers in *The Agile Times* in 2004: the risk burndown chart.

Before we create a risk burndown chart we first need something akin to the typical risk census, which is merely a list of a project’s top risks along with the probability of the risk occurring and the size of the loss that would occur if it did. An example is shown in the following table, which I’ve simplified by eliminating a few columns that are not essential to this post:

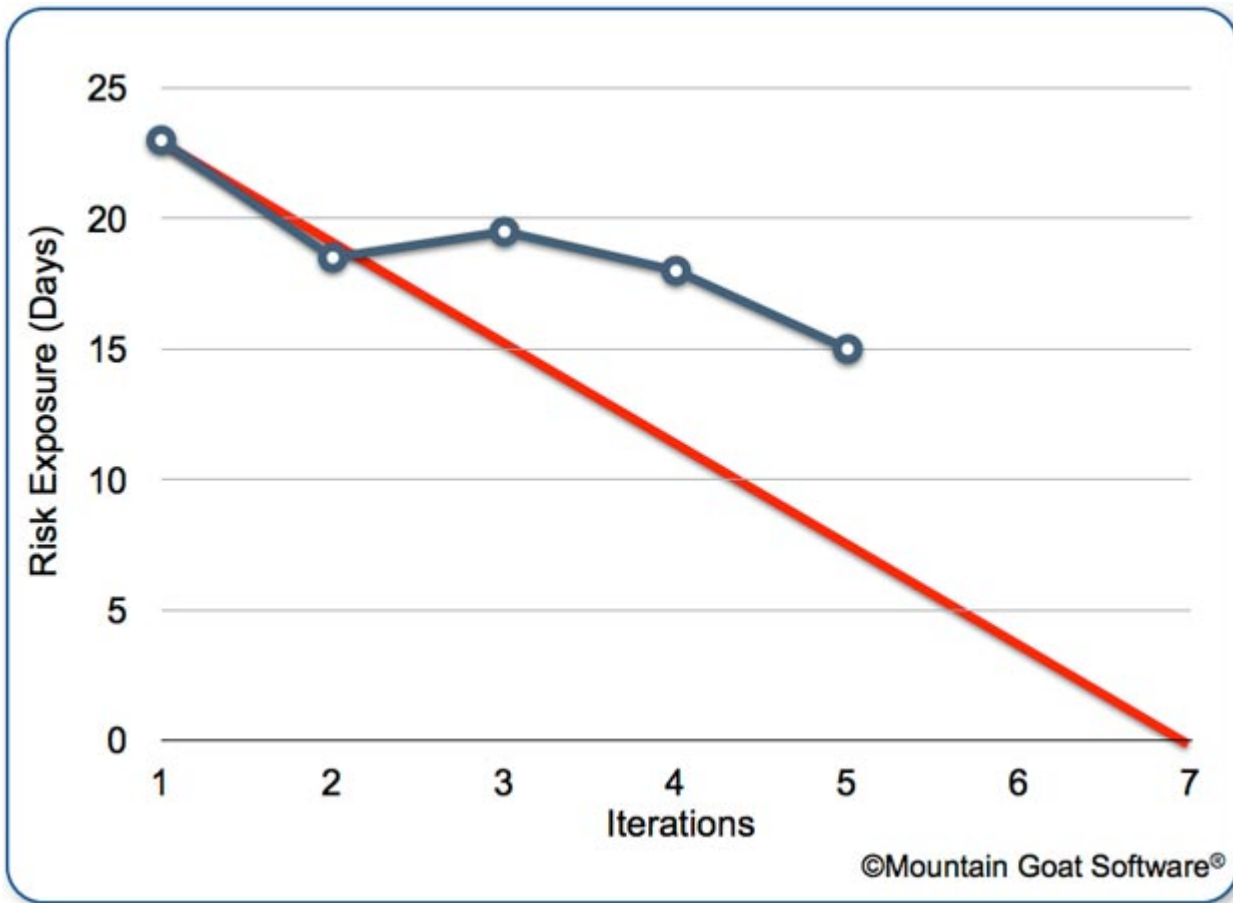
Risk	Probability of Risk	Size of Loss (Days)	Risk Exposure (Days)
Backup and restore may require the inclusion of additional third-party products.	20%	15	3
The lack of scientifically relevant sample data impacts Partner A's ability to validate the product.	35%	20	7
There won't be time for Partner A to provide feedback on the format of Analysis reports, which means they could find the reports unacceptable during validation.	10%	5	0.5
Partner A employees are not available to validate the new features until too late in the process, limiting our ability to make additional releases that address any issues they might uncover.	20%	5	1
There won't be time in the QA process to validate, equally, on all browsers on all operating systems.	40%	5	2
Partner A may require more end-user documentation than has been provided.	25%	20	5
Exposure:			18.5

Our simple risk census here describes each risk, provides an estimate of how likely the risk is to occur, the impact to the schedule if the risk did occur, and then the expected exposure to the risk which is the probability multiplied by the size of the loss. I recommend creating a risk census during the first sprint planning meeting and then updating it quickly during subsequent planning meetings as new risks are identified or as the probabilities or sizes of known risks change.

The risk burndown chart is then created by plotting the sum of the risk exposure values from the census. My recommendation is to sum only the top ten risks even if the team has identified more. Do this even if the top ten change over the course of the project. A sample risk burndown chart will look like this:



As with a regular release burndown chart, we should see a linear drop in risk over the course of the project, as shown in the red line of this next risk burndown chart.



When risk is not coming down at the appropriate rate, the team may want to budget some time in the next sprint to work directly on risk mitigation.

So as we can see from these examples, it is possible for risk management and [agile project management](#) to coexist on projects that need to explicitly do so. And the risk burndown chart provides a convenient way of visualizing changes in risk.

Posted: April 8, 2010

Tagged: teams, sprint planning, meetings, agile project management, burndown chart
