

Determining the Value of Ambiguous Agile Projects With Multiple Iterations Using Expected Commercial Value

Anita Friis Sommer

Agile Development for Physical New Products

Agile Development was created in the software industry through the 1990s, at a time when they needed new ways to overcome many deficiencies (1). By 2013, we began to see evidence of manufacturers of physical products – notably those with software embedded in their products – applying Agile to their *hardware developments*. Many other manufacturers were soon to follow (2,3,4,5).

While Agile provides many benefits to manufacturers, it also creates new challenges. One of these is how to place *an economic value on Agile projects*, which are characterized by high levels of ambiguity, uncertainty and fluidity. The article outlines the Expected Commercial Value (ECV) as a useful tool to gauge the economic value of highly ambiguous Agile projects with multiple iterations. The mathematical derivation of ECV equations for such projects is described in detail in the article's two appendices.

The author: Dr. Anita Friis Sommer is Senior Innovation Manager at LEGO Group and holds a post.doc from University of Cambridge on process modelling. Dr. Sommer is known for her research on Agile in combination with Stage-Gate.

She is PhD in Product Development Management and Civil Engineer in Industrial Management from Aalborg University, Denmark. Email: anita@sommersystems.com

How Agile-Scrum Works for Manufacturers

Most manufacturers already employ a gating or Stage-Gate system to drive new product projects from idea through to launch. When these firms adopt Agile methods (typically, the Scrum version of Agile), they keep their existing stages and gates, and merely build Agile into some or most of their existing stages, such as Development, Concept or even Ideation (6). The term “Agile-Stage-Gate hybrid model” has been coined to describe the combined model.

In practice, each stage of the process is broken into a series of short, iterative, incremental time-boxed *sprints*, each sprint typically about 2-4 weeks long. At the end of each sprint, the product is demo'd to the stakeholders – to management and sometimes to customers – for feedback and possible revision. Thus, the product definition evolves over the development phases of the product, rather than being fixed as in the traditional development process. Further, each sprint begins with a sprint planning meeting to decide what tasks will be undertaken during the next 2-4-week sprint. So the project plan also comes together as the project moves forward: There is no traditional plan for the next stage or for the entire project – no time-line, no Gantt chart, or no critical path plan – as found in traditional new-product projects.

Highly Ambiguous, Constantly Changing Projects

While Agile's flexibility and rapid response to new information has many benefits, this flexibility and fast-response has a negative side as well. The ambiguity of Agile-Stage-Gate projects makes it difficult to make effective Go/Kill decisions, especially at the early gates. With no fixed project plan at the project's beginning, it is almost impossible to accurately forecast people resources needed, their times, and the total duration of the project. Making a realistic *estimate of the likely development cost* as the project gets underway is therefore not

possible. Further, the product design evolves over time and goes through multiple changes via the iterations during the life of the project. Thus, arriving at an accurate estimate of *manufacturing costs*, *selling price*, *customer acceptance*, and even *likely sales* in the early stages of the project is not viable.

Major Challenge to Determining the Economic Value the Project

This ambiguous situation is a major challenge to both the project team, who must *create a business case* to justify their project, and to senior management who *must approve the project* and the *resources to undertake it*. How can one justify an investment in an R&D project – for example, undertake a financial analysis or profitability calculation – with unreliable and fluid estimates of sales volume and pricing, no firm idea of development costs and time, and only a vague idea of manufacturing costs and equipment needs! But there are solutions to decision-making in the face of extreme ambiguity.

Tools to Help Make Decisions Under Uncertainty

It's here where many of the project selection tools recommended when estimates of sales and costs are uncertain or fluid prove valuable. These include:

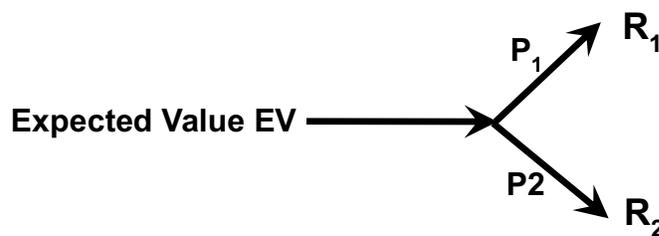
- Financial models, such as Options Pricing theory, ECV (Expected Commercial Value), probability-adjusted NPV, and Monte Carlo Simulation (7).
- Also scoring models which seem to work best with hard-to-quantify projects.

The ECV or Expected Commercial Value is particularly appropriate for highly uncertain and iterative Agile projects.

The Expected Commercial Value

A realistic financial model for handling the incremental or step-wise nature of investing in a new-product project is the Expected Commercial Value (ECV). The ECV looks at risks and probabilities, but most important, takes the investment a step-at-a-time via a *decision tree approach*. ECV simply computes the expected value of the outcomes of decisions made during the course of a new-product project.

Expected value is a familiar term, and is simply the sum of the consequences of two or more outcomes of a decision weighted by their probabilities of occurring:



In the sketch above, there are two possible outcomes of a decision, with consequences or results R_1 and R_2 . The probabilities of each outcome are P_1 and P_2 . Thus the *expected value* of the Go decision here (see sketch above) is simply:

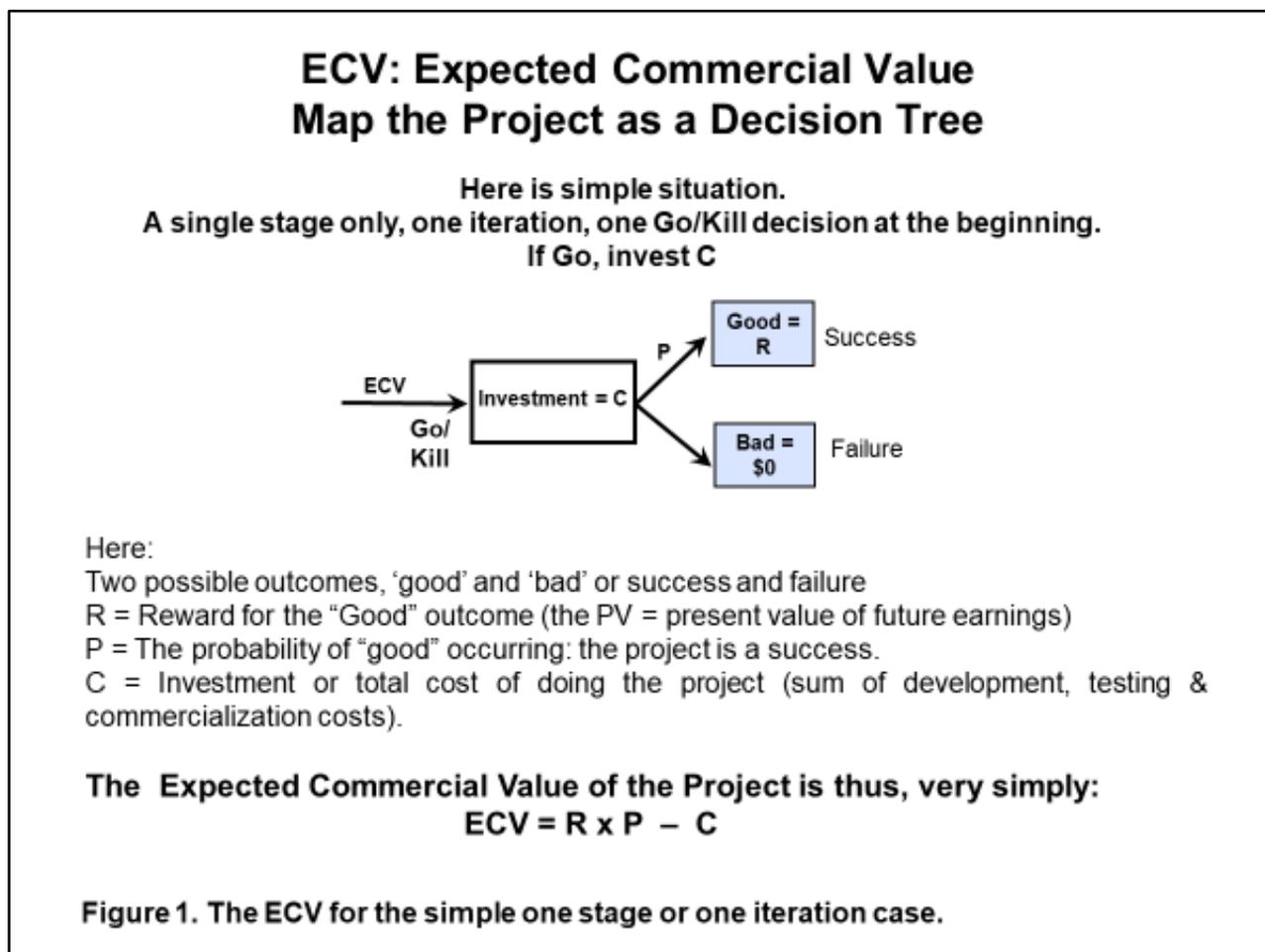
$$EV = P_1 \times R_1 + P_2 \times R_2$$

The ECV method is based on expected values, but looks at projects with multiple stages and thus multiple Go/Kill decisions, and uses a decision tree approach. The method was explained in a previous article for the general case of a 4-stage investment situation (8).

The ECV has particular applicability when the incremental investment decisions are much more frequent than merely four stages outlined in (8), and involve much smaller investments. That's the situation faced when using a sprint or iterative approach as in Agile-Stage-Gate. After any one of the many sprints, at the management demos for example, one can stop the project. But the decision to move forward is usually only for a sprint or two, and thus the investment amount is quite small, and risk is mitigated.

The Simplest Case: A Single Stage or One Iteration Model

How does ECV work for Agile projects and what impact does it have on risk? The simplest case is a single stage model with two possible outcomes, "good" and "bad" or "success" and "failure" – see Figure 1.



Here the ECV is simply the expected value of the outcome, namely:

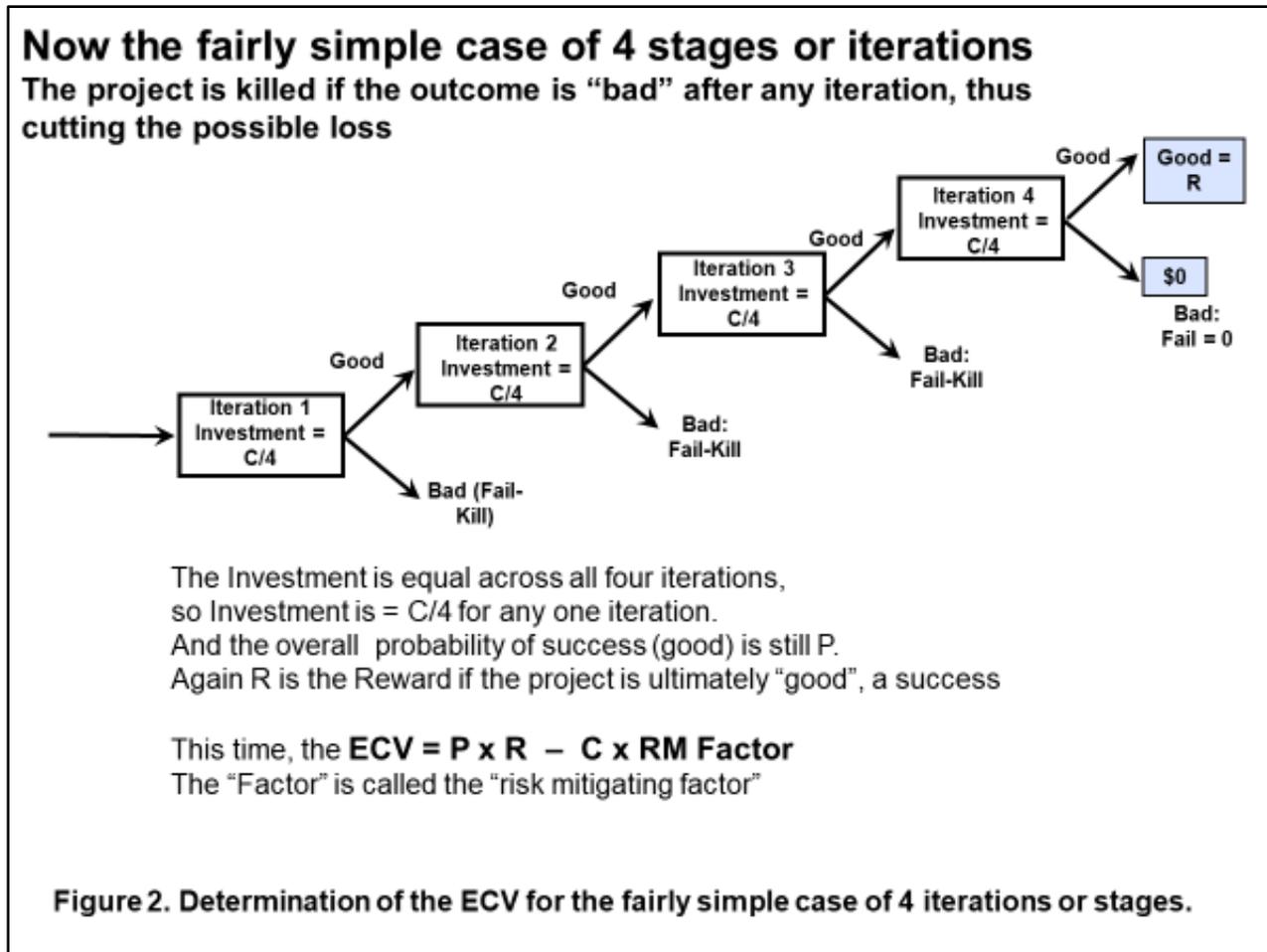
$$ECV = R \times P - C$$

where R is the reward or payoff from the project (usually the present value of future earnings); C is the cost of undertaking the project (development and commercialization costs); and P is the probability of success.

Note that the traditional NPV would be: $NPV = R - C$; so even for this simplest case, the ECV gives a more realistic result by including the probability of success.

A Four Iteration Project

The situation with of four iterations – where spending decisions (Go/Kill) can be made between (or after) each sprint or iteration – is shown in Figure 2. Every iteration has an outcome, “good” or “bad”; if the result is bad, the project is stopped or killed with zero consequences.



Again, R is the Reward if the project is ultimately “good”, a success. And the consequence is zero if the project is stopped or killed. The Investment, C , is equal across all four iterations, so the Investment is = $C/4$ for any one iteration.

The overall probability of success (“good” outcome) is still P , namely the probability of success for the entire project. Therefore the “probability of good” in any one iteration is $P^{1/4}$. This means that when the four probabilities are multiplied together – the chain of the probabilities to yield the overall probability – the result is simply:

$$\text{Overall probability (the chain)} = P^{1/4} \times P^{1/4} \times P^{1/4} \times P^{1/4} \text{ which equals } P.$$

This time from Figure 2, the ECV works out to be:

$$\text{ECV} = P \times R - C \times \text{RM Factor}$$

The derivation of this equation is in Appendix A.

The “Factor” in Figure 2 can be shown to be:

$$\text{RM-Factor} = 1/4 \times (1-P) / (1- P^{1/4})$$

This RM-Factor is a number from zero to one, and thus is called the “risk mitigating factor, because it, in effect, reduces the value of the investment, thus reducing risk. The RM-Factor actually ranges from 0.73 to 0.88 as P goes from 40% to 70% (40-70% is used as a range of success probabilities for a typical risky project when starting out). See Appendix A.

Thus, by having four instead of one iteration, the project increases in value between 12% and 27% of C. That is, the downside risk of losing C is partially mitigated.

N Stages or Iterations

The situation with multiple or N iterations is similar to the 4-iteration case, but is the “general case” shown in Figure 3. For an Agile-Stage-Gate project, N is very large – that is, many but small iterations or short sprints where spending decisions (Go/Kill) can be made between (or after) each sprint or iteration.

Only a few iterations are shown here in Figure 3. As in the 4-iteration case in Figure 2, every iteration has an outcome, “good” or “bad”; if the result is bad, the project is stopped or killed with zero consequence.

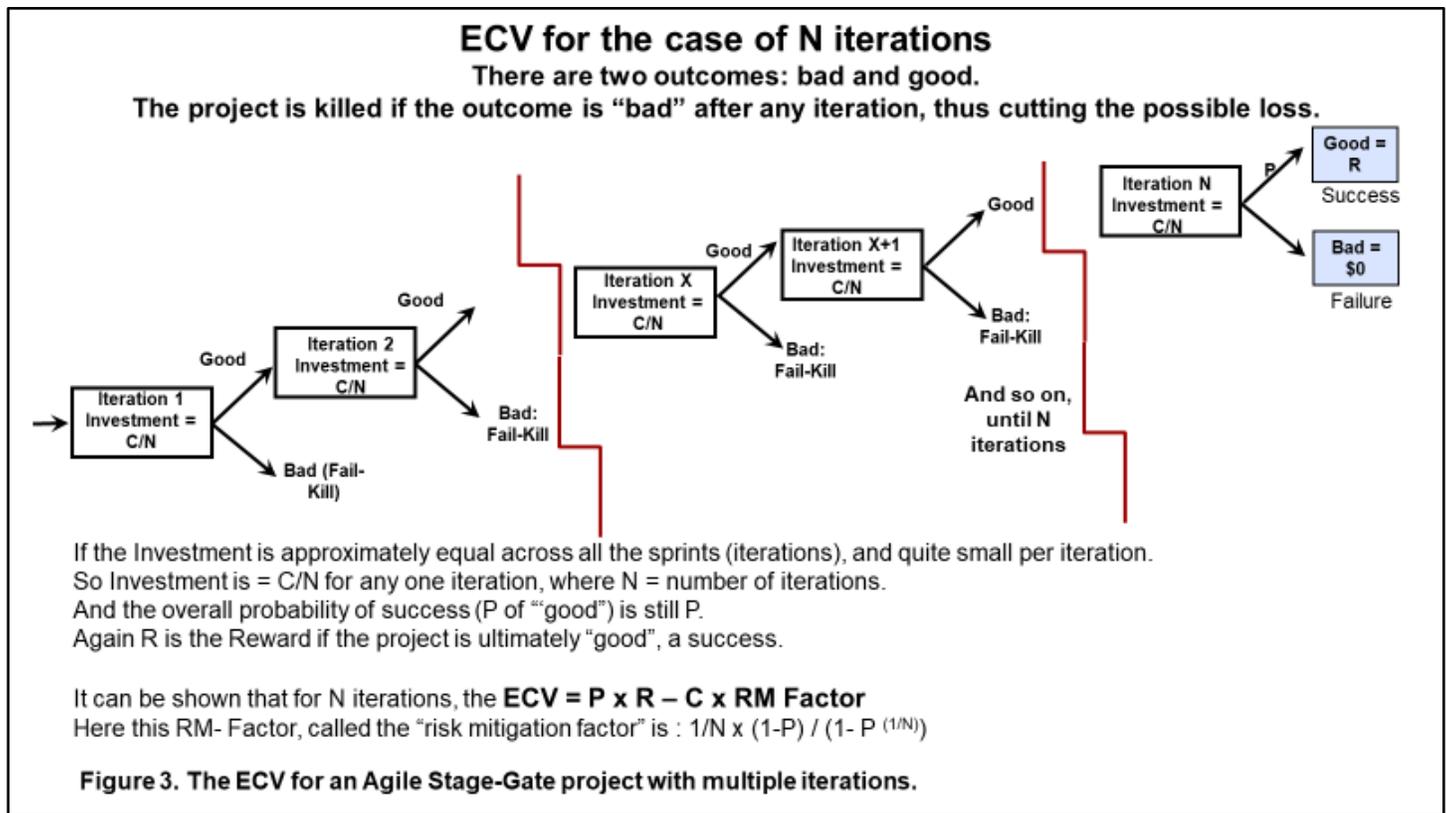
The overall probability of success (“good”) is still P, the probability of success for the entire project. Therefore the “probability of good” in any one iteration is $P^{1/N}$. This means that when the N probabilities are multiplied together – the chain of the probabilities to get the overall probability – the result is simply:

$$\text{Overall probability (the chain)} = P^{1/N} \times P^{1/N} \times P^{1/N} \times P^{1/N} \dots N \text{ times which equals } P.$$

For the situation with N iterations, the ECV can be shown in Appendix B to be the same as for the four-iteration case, and even similar to the single stage case, namely:

$$\text{ECV} = R \times P - C \times \text{RM Factor}$$

Again, R is the reward and P the probability of success. And once more, the factor in the ECV equation is called the “risk mitigating factor”, which again is a number between zero and one. Thus, having multiple iterations has the effect of *reducing the size of the investment* and thereby mitigates risk. Appendix B shows the details of the derivation of the ECV equation and the RM-Factor for the N-iteration situation.



When there are many iterations, as in Agile-Stage-Gate, this RM-Factor ranges from 0.66 to 0.84 when the probability of success, P, is between 40% and 70%, which, as noted above, is a fairly typical range for riskier projects. Thus, in Agile project, with many iterations, the project increases in value by between 16% and 34% of the investment, C; that is. the *downside risk of losing C dollars is significantly reduced.*

In practice, the project team can use the general equation in Figure 3 to arrive at on-going estimates of the ECV as the project moves along. Alternately, the team can create their own model or decision tree diagram, much like Figure 3, but with iterations, probabilities, and outcomes that reflect their own project. Note that the calculations will be similar – a series of expected value calculations – in order to calculate the ECV..

Estimating Probabilities of Success

One added complexity with ECV is the *need to estimate probabilities of success*, both technical and commercial. By using a modified Delphi method, however, the project team plus a few invited subject-matter experts in a facilitated group discussion, and using prompting questions, can usually come to a realistic estimate after a few rounds (9 Ch 8, 10)

There are other recommended methods, such as using scorecards to estimate the likelihood of commercial and technical success (7). And some firms simply keep historical records of project outcomes and thus have developed handbook-like tables with these probabilities given for different types of projects. But the modified Delphi is likely the most universally applicable and straightforward to use, and besides, yields good results.

Implications for Agile and Ambiguous Projects

The ECV and its associated equations have a number of benefits when evaluating Agile-Stage-Gate projects; and they also provide important insights to the economic value of such ambiguous and iterative projects.

Deals with project ambiguity appropriately: Agile projects, which are often ambiguous near their beginning, have uncertain outcomes, both technical and marketing: There are many unknowns in the early stages of the project. The project team can often make estimates of these probabilities of success; and although just estimates, these probabilities can be used to help place a *more realistic economic value* on the project. The ECV method builds in these probability estimates in an appropriate and effective way. Further, any traditional evaluation method, such as NPV or Payback Period, which ignores these probabilities, in essence assumes a 100% chance of success, and thereby is inherently invalid.

Allows for killing of projects: The ECV is built on the premise that not all projects will go all the way through to launch. One of the outcomes at the end of every iteration is a possible “kill” decision to terminate the project and stop spending any more money. Thus *this “kill option” is built in by design*, and the ECV calculations indeed do account for the kill outcome: Investment is cut and thus saved in the case of a “bad” project. By contrast, NPV is a capital budgeting technique, and assumes that the entire investment is made – an “all or nothing” decision – which is invalid in the case of an iteratively-undertaken project.

Risk reduction: Multiple iterations help to mitigate the risk of the project by, in effect, reducing the impact of potential downside losses. As shown in Figure 3, many iterations effectively reduce the potential downside loss (reduce the value of C in Figure 3) by a risk mitigating factor; that is, the *effective investment is reduced by somewhere between 16% and 34%*, which can have a significant impact on the economic value of the project.

Adjusting the probabilities along the way: As more information becomes available with successive iterations, the probabilities of success can be adjusted in the ECV. As the likely technical solution becomes more evident, and as market testing results provide positive data, the hope is that the probabilities of both technical and market success begin to increase over the project’s many iterations. The ECV calculation allows for this, and indeed, the ECV can be recalculated in real time, with adjustments based on new knowledge to reflect the value of the project as it moves forward. Note however, that if these probabilities start to trend downward over time, a recalculated ECV might begin to signal a kill decision. Thus the ECV becomes *a useful tracking metric to gauge the economic health* of the Agile project in real time.

Weighing the potential upside gain versus the downside loss at each decision point: Many companies use the upside gain versus the potential downside loss as a measure of risk-return when looking at Go/Kill decisions. When considering upside gains, the decision-makers should look at the real-time value of the ECV; but in so doing, they should compare that ECV, not to the total investment C, but to *investment required for the next few sprints*. Thus, the use of multiple and smaller iterations greatly improves this risk-return ratio.

Used to prioritize projects: Traditionally the NPV was used to calculate the Productivity Index to help in project prioritization (8). The Productivity Index (PI) measures the value to the company for each additional person-day (or dollar) spent on the project. But for ambiguous and Agile projects, the ECV is the clearly the more appropriate value to use in calculating the PI (instead of the NPV):

$$PI_{A-SG} = ECV / \text{Work Yet To Be Done in Project (work-days)} \quad \text{or}$$

$$PI_{A-SG} = ECV / \text{Investment Yet To Made}$$

The Productivity Index, based on the ECV, can thus be a very useful tool when trying to decide which projects should receive the greatest priority in a portfolio management decision. Think of this “new” index with ECV as a probability-adjusted Productivity Index, which gauges the “bank for buck” of the project..

Conclusion

The ECV is a powerful tool for determining the economic value of a new-product project under conditions of uncertainty and ambiguity, conditions which are found in an Agile-Stage-Gate project. Further, risk can be significantly mitigated by building in multiple iterations with the possibility of killing the project after any one iteration. The mathematical derivations of a 4-iteration and the more general N-iteration case were outlined above and in the two appendices. The benefits of the ECV method when used in a multi-iteration project along with the insights gained, are many. In practice, a project team, rather than using the general model, might prefer to develop their own decision mode to capture the nuances of their own project – a model similar to the ones presented here – and thus calculate their ECV specific to their project.

* * *

Appendix A: Derivation of ECV Equation for the Four Iteration Case

The situation is described in Figure 2, shown again below. Starting on the right in Figure 2, with Iteration #4:

$$\text{Expected Value} = R \times P^{1/4} - C/4$$

Note: C is spent equally across the 4 iterations, thus investment is C/4 per iteration.

Moving one iteration to the left, to Iteration #3:

$$\text{Expected Value} = [R \times P^{1/4} - C/4] \times P^{1/4} - C/4$$

Moving left again, this time to Iteration #2:

$$\text{Expected Value} = \{[R \times P^{1/4} - C/4] \times P^{1/4} - C/4\} \times P^{1/4} - C/4$$

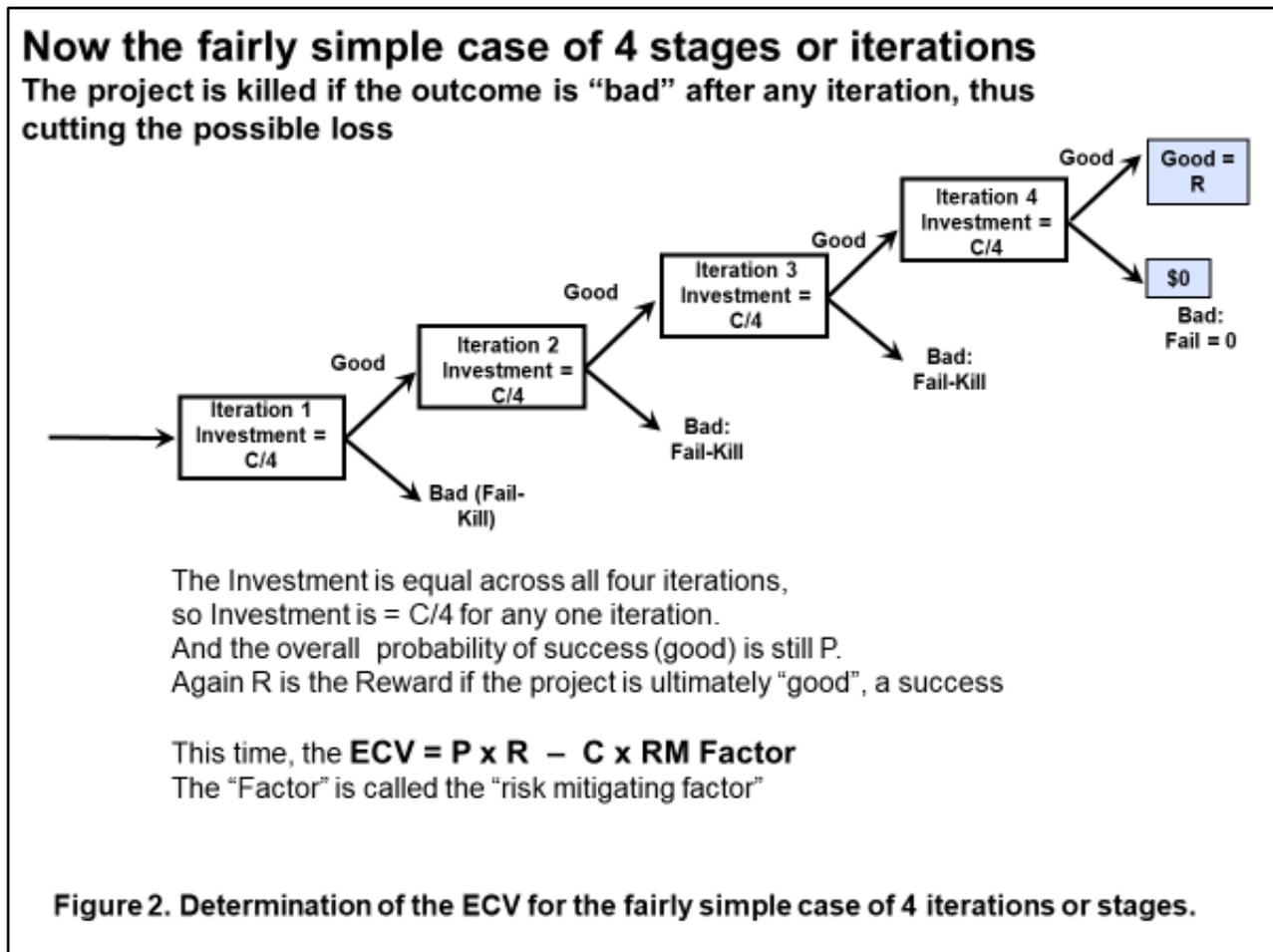
And finally, moving left to iteration #1:

The total project value, or Expected Commercial, ECV is:

$$\text{ECV} = (\{[R \times P^{1/4} - C/4] \times P^{1/4} - C/4\} \times P^{1/4} - C/4) \times P^{1/4} - C/4 \quad (1a)$$

Multiplying Equation (1a) through, and factoring out the C/4 term, gives (1b):

$$\text{ECV} = R \times P - C/4 \times [P^{3/4} + P^{2/4} + P^{1/4} + 1] \quad (1b)$$



Now Equation (2) is developed to replace the sum of the $P^{X/4}$ terms in (1b).

$$\text{Sum} = S = P^{3/4} + P^{2/4} + P^{1/4} + 1 \quad (2)$$

So Equation (1b) becomes:

$$\text{ECV} = R \times P - C/4 \times S \quad (1c)$$

Now the value of S is determined:

Multiply the series in Equation (2) by $P^{1/4}$ to get (3) :

$$S \times P^{1/4} = P + P^{3/4} + P^{2/4} + P^{1/4} \quad (3)$$

And subtract the two equations, (3) – (2) to get:

$$S \times (P^{1/4} - 1) = P - 1 \quad (4)$$

$$\text{Thus } S = (P - 1) / (P^{1/4} - 1) \quad (5)$$

And so substituting Equation (5) into (1c) gives, for 4 iterations:

$$\text{ECV} = R \times P - C/4 \times [(P - 1) / (P^{1/4} - 1)] \quad (6)$$

Call the term $\frac{1}{4} \times [(P - 1) / (P^{1/4} - 1)]$ the “Risk Mitigating Factor” or simply “RM-Factor” for short.

$$\text{Thus } \text{ECV} = R \times P - C \times \text{RM-Factor} \quad (7)$$

which is very similar to the simple one-iteration case in Figure 1, except C is adjusted by the Risk Mitigating Factor, where $\text{RM-Factor} = \frac{1}{4} \times [(P - 1) / (P^{1/4} - 1)]$

Using various trial values of P between zero and 1.0 shows that the RM-Factor is a number also between 0 and 1.0.

Appendix B: Derivation of ECV Equation for the General Case of N Iterations

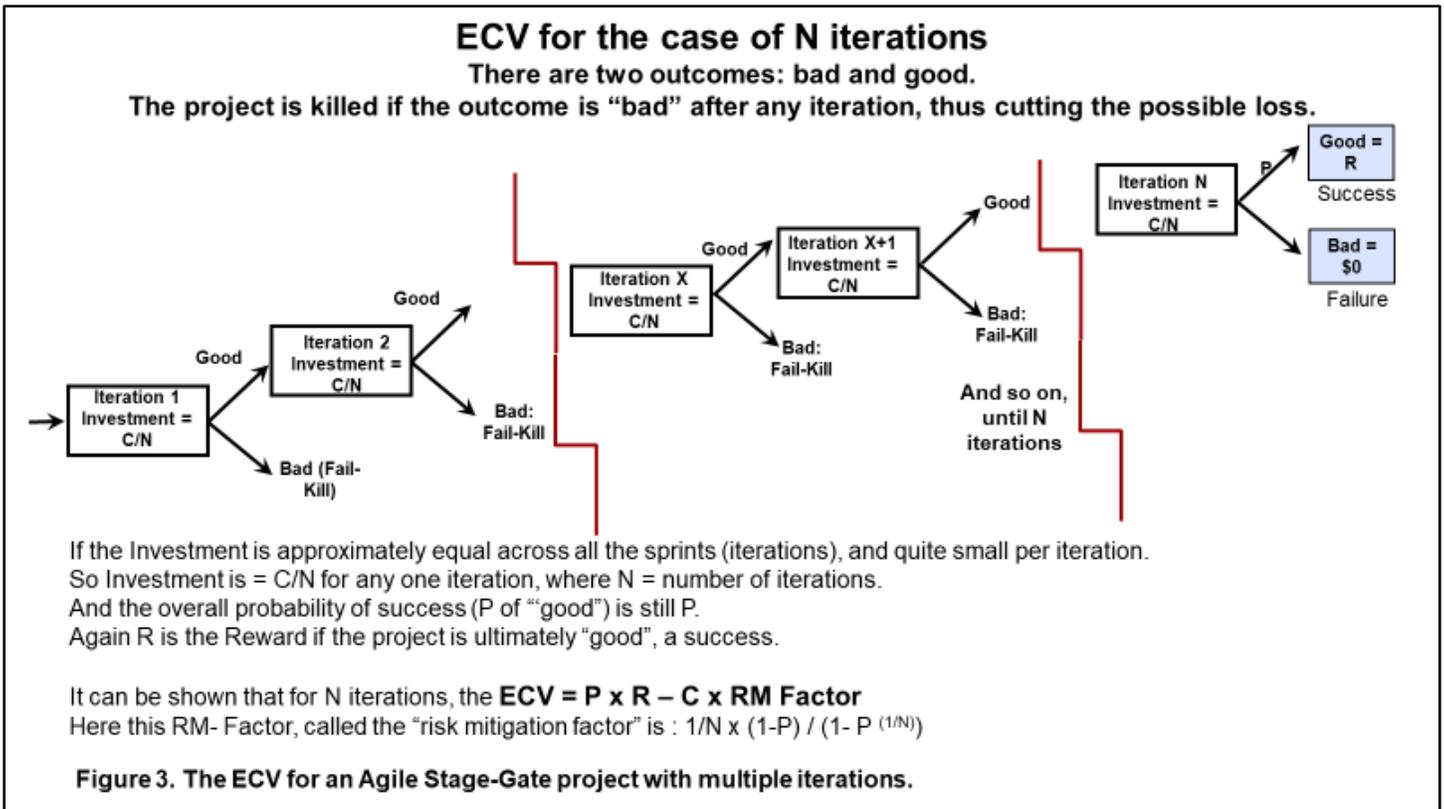
This derivation is much the same as the four-iteration case above, except for N iterations.

Starting once again on the right in Figure 3, shown below again, with Iteration N, and working to the left:

$$ECV = (\{ [R \times P^{1/N} - C/N] \times P^{1/N} - C/N \} \times P^{1/N} - C/N) \times P^{1/N} - C/N \times \text{etc. to N terms} \quad (8a)$$

Multiplying Equation (8a) through and factoring out C/N as before gives:

$$ECV = R \times P - C/N \times [P^{(N-1)/N} + P^{(N-2)/N} + P^{(N-3)/N} + \dots + P^{1/N} + 1] \quad (8b)$$



Now, as before, Equation (9) is developed to replace the sum of the $P^{X/N}$ terms

$$\text{Sum} = S = P^{(N-1)/N} + P^{(N-2)/N} + P^{(N-3)/N} + \dots + P^{1/N} + 1 \quad (9)$$

So Equation (8b) becomes

$$ECV = R \times P - C/N \times S \quad (8c)$$

Again the value of S is determined:

Multiply the series in Equation (9) by $P^{1/N}$ to get (10) :

$$S \times P^{1/N} = P^{N/N} - P^{(N-1)/N} + P^{(N-2)/N} + \dots + P^{1/N} \quad (10)$$

And subtract the two equations, (10) - (9) to get:

$$S \times (P^{1/N} - 1) = P^{N/N} - 1 = P - 1 \quad (11)$$

$$\text{Thus } S = (P - 1) / (P^{1/N} - 1) \quad (12)$$

which is simply the general equation for S (and also explains the 4-iteration version when N=4, namely Equation (5)).

Substituting Equation (12) into (8c) gives, for N iterations:

$$ECV = R \times P - C/N \times [(P - 1) / (P^{1/N} - 1)] \tag{13}$$

Once again, call the term $1/N \times [(P - 1) / (P^{1/N} - 1)]$ the “Risk Mitigating Factor”.

$$\text{Thus } ECV = R \times P - C \times \text{RM Factor} \tag{14}$$

$$\text{where RM-Factor} = 1/N \times [(P - 1) / (P^{1/N} - 1)] \tag{15}$$

Note that Equation (14) is the same as for the simpler 4-iteration case in Figure 2 and Equation (7), with C adjusted by a more general Risk Mitigating Factor for N iterations.

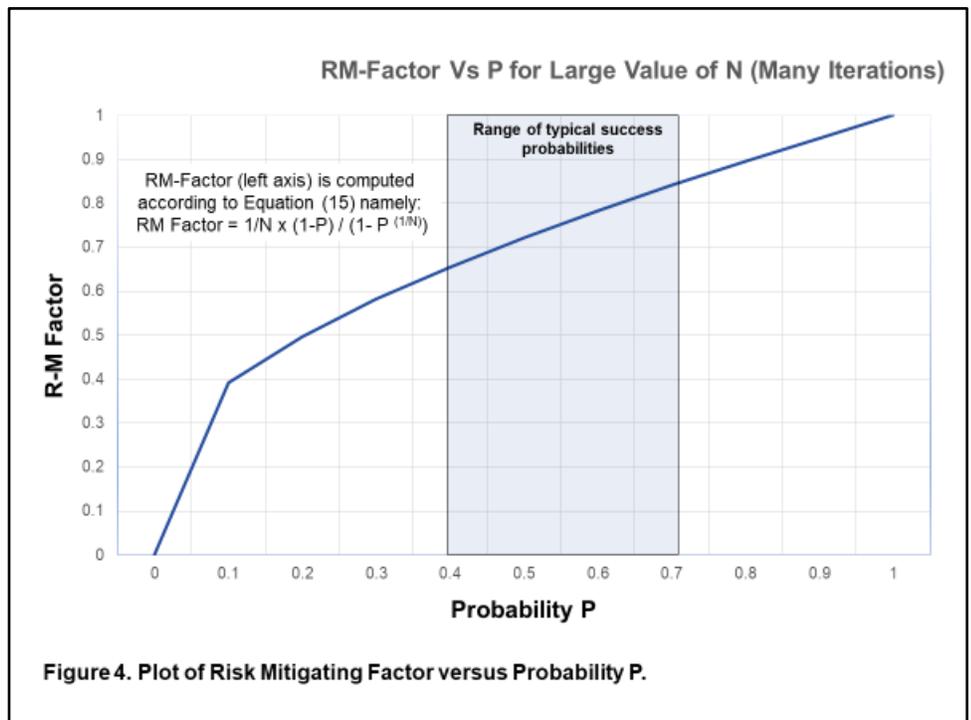
Again, using various trial values of P between zero and 1.0 shows that the RM-Factor is also a number between 0 and 1.0.

Summing up, the general N-iteration equation is Equation (14):

$$ECV = R \times P - C \times \text{RM Factor} \tag{14}$$

$$\text{where RM-Factor} = 1/N \times [(P - 1) / (P^{1/N} - 1)] \tag{15}$$

Again, a range of 40-70% is used for P, the overall success probability for a typical risky project when starting out, which yields a RM-Factor ranging from 0.66 to 0.84. Figure 4 shows a plot of the Risk Mitigating Factor calculated from values of P, with the 40%-70% probability range shaded. Thus, by having N iterations instead of one, the project increases in value between 16% and 34% of C. That is, the downside risk of losing C is partially mitigated.



References:

- ¹ Cooper, R.G. “Next in New-Product Development: Agile-Stage-Gate Hybrids,” *CIMS Innovation Management Report* Nov-Dec 2016, pp. 10–14.
- ² Cooper, R.G., “Agile-Stage-Gate Hybrids: The Next Stage For Product Development,” *Research-Technology Management* Jan 2016, vol. 59, no. 1, pp. 1–9.
- ³ Cooper, R. G., and Sommer, A. F. “The Agile-Stage-Gate hybrid model: A Promising New Approach and a New Research Opportunity,” *Journal of Product Innovation Management* 2016, vol. 33, no 5, pp. 513–526.
- ⁴ Cooper, R.G., and Sommer, A.F. “Agile-Stage-Gate: New Idea-to-Launch Method For Manufactured New Products Is Faster, More Responsive,” *Industrial Marketing Management* Nov 2016, vol. 59, pp. 167–180.
- ⁵ Cooper, R.G., and Sommer, A.F. Agile-Stage-Gate for Manufacturers – Changing the Way New Products Are Developed”, *Research-Technology Management* Mar-Apr, 2018, vol. 61, no. 2, pp. 17-26.
- ⁶ Invention Center 2018. Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), *Consortium Benchmarking 2018 “Agile Invention”: Evaluation of the Study Results*.
- ⁷ Cooper, R.G. 2013. Where are all the breakthrough new products? Using portfolio management to boost innovation. *Research-Technology Management* 156(5): 25-32.
- ⁸ Ibid, 2013.
- ⁹ Cooper, R.G., Edgett, S.J. and Kleinschmidt, E.J. 2002. *Portfolio Management for New Products*, 2nd edition. New York, NY: Perseus Publishing.
- ¹⁰ Delphi Method. As described in Wikipedia. https://en.wikipedia.org/wiki/Delphi_method