The Aptage Release Planner

Dr. Murray Cantor, CTO Managed Innovation Systems

Managed Innovation Systems
https://aptage.com

Introduction

The Aptage Release Planner (ARP) is an agile planning and tracking tool available as a web service at www.aptage.com. This paper describes the value ARP provides, and how to use it.

Why ARP

Often businesses have what are, to some, contradictory development team goals:

1) Creating delivery plans that can be counted on to support business plans
2) Employing agile methods for improving development efficiency

The key mechanism for achieving these goals of coordinating the needs of the business with the delivery capability of an agile team is release planning: determining what features or epics can be planned for the next release. However, release planning is especially challenging for Agile teams since both the size of the effort and the team velocity have some degree of uncertainty. ARP is designed to facilitate release planning while fully accounting for these uncertainties.

Agile teams plan releases in terms of both large and small chunks of functionality. Large chunks are usually called *features* or *epics*. Features are used in product or system planning to specify, in broad terms, what the system or product does. Some examples of features include:

- Online check depositing is a feature of a mobile bank portal
- Printing to .pdf format is a feature of a word processor
- Displaying usage analytics is a feature of a webhosting system

At the highest level, an Agile release plan is the set of features, either partial or complete, that will be delivered on a certain date. Often the business needs the development team's assurance that the planned features will be available in order to support other business plans, such as marketing activities, support training, revenue planning, or meeting customer commitments.

Small chunks of functionality are called *stories*. A story

1) Describes some small piece of functionality from the users' perspective
2) Is containable in a sprint, usually by a single programmer

In Agile, features or epics are decomposed into stories. Planning to deliver some of the functionality of a feature entails specifying a subset of that feature's stories that deliver the functionality. Therefore, a high-level release plan consists of a set of features, and a detailed Agile release plan consists of the features, and, for each feature, the included stories. The release planning challenge then is to find plans that are neither:

- Too aggressive – so large that it is unlikely the team can deliver as planned, or
- Too conservative – so small that the team is not delivering sufficient value to the business.

ARP's value is helping management and the team agree on a 'just right' plan.
Any release plan can be characterized as a sort of bet made by the development team and the management. They are both betting that the team will deliver the planned content on the planned date. They can increase the odds of winning by committing to minimal content, but this would result in the team delivering less value than it might. On the other hand, committing too much content, perhaps under stakeholder pressure, can negatively impact the business with many missed deadlines. The challenge is to find the right amount of content so that the plan is neither too conservative nor too aggressive.
What follows describes how ARP uses Agile planning methods to compute the odds of winning the bet and finding the 'just right' plan. This begins with a discussion of uncertainty.

Uncertainty, Innovation and Learning

Often, the role of software and IT departments is to deliver and maintain novel, innovative solutions to the business and/or to the market. By definition, novel things are new and, consequently not completely understood. To deliver anything novel requires experimentation, false starts, and learning. At the onset of a novel effort the team has much to learn. The more novel the effort, the more the team has to learn. It stands to reason that a team is unable to make predictions with certainty for novel effort – they simply do not know enough. Asking the team to commit to a firm, fixed delivery plan for a novel effort is unreasonable and inconsistent with Agile principles and practices. A key insight behind ARP is that the certainty of estimates will vary with the team's knowledge of what it takes to deliver. This relationship between novelty and uncertainty raises a dilemma for managing a development team. The stakeholders need both:

- **Novelty** – creative solutions to business problems or market pressures
- **Certainty** – the ability to make business plans based on the availability of the code

But you cannot have both! Agile release planning is not an oxymoron, but a way to find the balance between the novelty and certainty. ARP is sensitive to team's learning and

provides the means for both sides to see the odds of winning the bet and agreeing on the 'just right' balance.

Eliciting Uncertain Quantities

As discussed above, asking the team for a precise estimate on the time or effort it will take to do something novel is not sensible. However, the team members do have some idea of the sizing. ARP leverages the ideas of Douglas Hubbard[1] by asking the estimator not for a number, but rather the triplet: {low guess, likely guess, high guess} of the sizing.

For example, suppose a team lead and a developer are having a discussion on how long a task would take. The conversation might go like this:

- *Lead asks, "How long it take to do the work?"*
- *Developer says, "Beats me, I have never done this before."*
- *Lead then might say, "Well, could you get it done in a year?"*
- *If Developer is being constructive and trusting they might say, "I know I could get it done in a year, probably sooner."*
- *Lead: "Great, what is longest it might take?"*
- *Developer: "Let's say six months.*
- *Lead: "Okay, if all goes well, what is the least time?"*
- *Developer; "I would say three months."*
- *Lead: "Fine, and what is your best guess?*
- *Developer: "My gut feeling is four months, but I will have a better estimate after I find out a few things."*

The lead now has the best possible information. She now knows it is very unlikely to be done in under three months, it is a very likely that it will be available within six months and the most likely sizing at four months. The triplet {3 months, 4 months, 6 months} is better information than any single number. It guides the lead how to place her bets and what sort of risks she is assuming. Furthermore, the conversation could be followed up with the following discussion:

- *Lead says: "That is a pretty wide estimate, what would it take for you to be more certain? What information would you need?"*
- *Developer might say: "We are using some technology that we have never used before. Once we have used it, I will know better how much effort is required."*
- *Lead then says, "Let's start immediately experimenting with the technology and revisit the estimate when you know more."*

[1] Douglas Hubbard, *How to Measure Anything: Finding the Intangibles in Business (3rd Edition)*, Wiley 2014.

Using ARP

ARP is a web service at www.aptage.com. Briefly, the user uploads an Excel workbook and within seconds, a pdf report is returned. In the next sections, we describe how to populate the input workbook and how to interpret the resulting reports.

ARP input

The input template is available at the website. It consists of two sheets. The first, the backlog sheet, is the content of the release, sized in story points as triples (low, expected, high).

Agile teams plan releases in terms of backlog items. In agile, these are sized in story points. The rate the team completes the backlog items is called the team velocity and is measured in story points per period. Hence the quotient of the planned number of story points by the velocity is the planned duration of the plan.

Both the story estimates and the velocity are uncertain. Because the teams velocity is uncertain it to should be specified as triplets, {low, expected, high}.

There are two cases of backlog items:

1) **Decomposed –** The feature is decomposed into stories, with each story sizing as a triplet {low, middle, high}. For example, the input for a planning poker 8-point story would be {5, 8, 13}. BAP is not restricted to the planning poker triplets. For example, the team could enter {10, 15, 15}. Additionally, if the team is entirely certain of the size they can use the same number for each value of the triplet. For example, if all agree the story size is 8, they can specify {8, 8, 8}

2) **Not Decomposed –** If it early in the planning cycle, a feature may be under consideration for inclusion, but not yet decomposed. In this case, the team provides the triplet for the overall feature. For example, a feature size could be the triplet {70,90,120}.

Note that eliciting the story points as triplets is a small modification of planning poker, a standard Agile practice[2]. In planning poker, each member estimates the size of the story using planning poker cards that are valued using modified Fibonacci numbers (1, 2, 3, 5, 8, 13, 21, 40…). Each team, perhaps after some discussion, settles on the middle value of the choices. So for a given story, if some team members choose 5 story points, some 8, and some 13, the team will settle on 8 story points. If there are more extreme choices, they are ignored. Note that even though the team has settled on a value, the team is uncertain overall as to the size of the story. Rather than ignoring this uncertainty, the team can use the range of values they came up with to specify the triplet.

[2] See, for example, https://en.wikipedia.org/wiki/Planning_poker

It is important to note that providing the triplets is generally less work than standard planning poker. The team can agree on the triplets more readily than the single value.

For each planning scenario, the user enters the data into an Excel workbook. The first sheet is a list of the backlog items and their size triplets. For convenience, there is also an inclusion flag: A '1' means the item is included in the plan. This makes it easy to develop plan alternatives and to track ongoing plans, moving the flag to '0' when the item is completed or now out of plan.

Backlog Item	Included (1 or 0)?	Low	Medium	High
1A	0	3	5	8
1B	0	5	8	13
1C	0	13	21	30
2	1	16	30	35
3	1	20	23	50
4A	1	21	30	40
4B	1	16	30	35
5A	0	20	23	50
5B	0	21	30	40

Figure 1. The ARP backlog sheet

The second sheet (Figure 2), has the project data:

- An estimate of the team's velocity in story points/per periods., also provided as a triplet {best case, likely case, worse case}. For example, a team expects to deliver about 40 story points per sprint, but believe that they might be able to do as much as 60, but at a minimum they can be counted on delivering 25. In this case velocity would be entered as the triplet {25, 40, 60}.
- The number of periods in the current plan.

Period Team Velocity Low	Period Team Velocity Probable	Period Team Velocity High	Remaining Release Duration in Periods
20	30	40	2

Figure 2 The ARP project data sheet

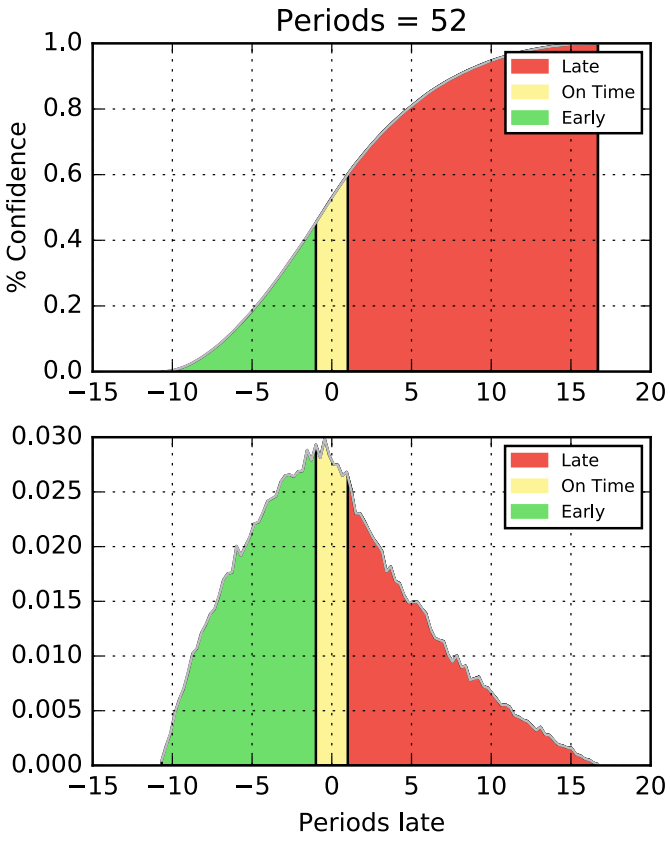
After, one can then use ARP to generate the likelihood of successful completion for different scenarios, combinations of features and, if entered, stories per feature. Figure 7 shows three examples of different scenarios.

ARP Output

Reasoning about bets entails probabilities. As discussed in the following section, ARP uses the team's best understanding of the effort sizings and team velocity to give the probability of delivering on time. The results are given in three views (Figure 3).

- The confidence diagram
- The probability view
- The betting table

These are each described below.



	Poor Bet	Even Bet	Good Bet
Periods Late	-3.9	-0.4	3.6

File: baptest.xlsx,
Timestamp: 2016-04-15 10:04:07

Figure 3. The ARP report

The red/green diagram

The red/green diagram (figure 4) shows the range of outcomes for the release effort, along with the probability of the outcomes. The outcomes are described in terms of how early or late the release is expected to be. The X-axis' zero point is the target date; negative values are weeks prior to the target date (early), and positive values are weeks after the target date (late). Early dates are shown as green areas in the chart; late dates are shaded in red.

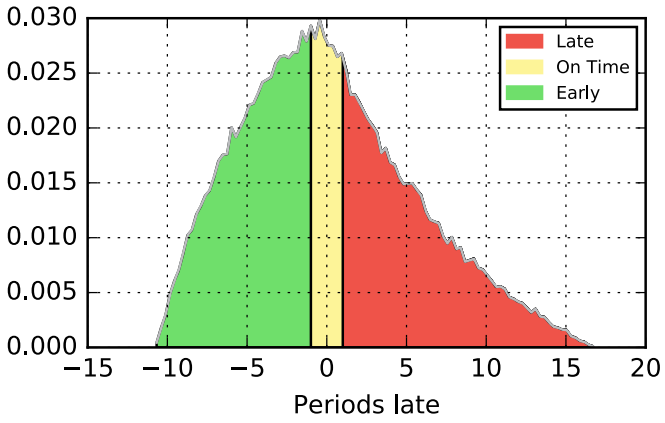


Figure 4. The red/green diagram

The ratio of red to green that makes up a 'just right' plan depends on your organization's risk appetite.
The reasoning behind a plan's mixture of red and green is supported by the betting table.

The Confidence Diagram

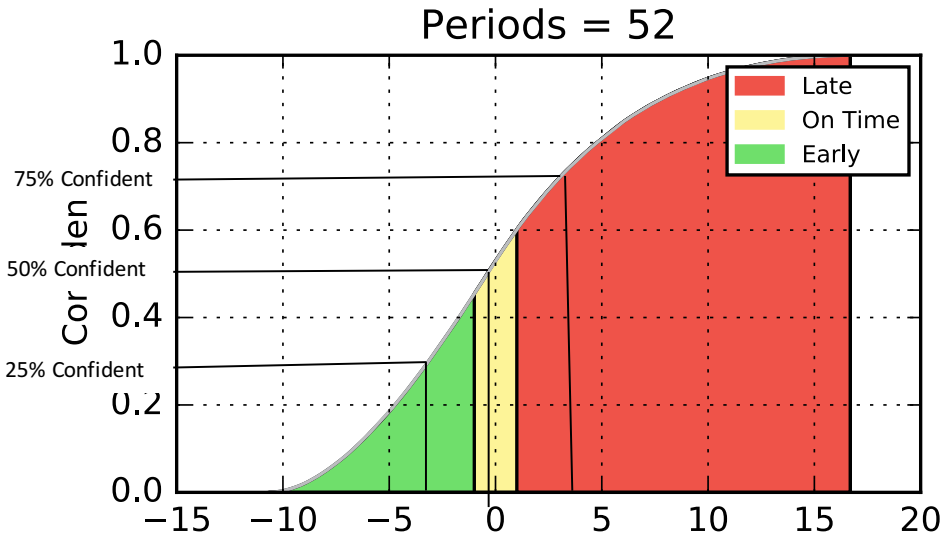


Figure 5. The ARP confidence diagram

This diagram provides a different view of the red/green diagram. In this diagram, the y-axis is the confidence in the ability to deliver. For example, in Figure 5. based on the

input the team should be 75% confident they can be no more than 3.6 late, 50% confident they will be on time, and 25% confident they could be about 4 weeks early.

The Betting Table

	Poor Bet	Even Bet	Good Bet
Periods Late	-3.9	-0.4	3.6

Figure 6. The ARP betting table.

ARP output also includes a betting perspective on the plan (Figure 6). If you were to bet on *when* you were to deliver on the planned content, what would be a good bet? The betting table provides a view of this from an odds-maker's perspective.
If you were an odds-maker, you would set the odds as:

- Winning the bet that the team will deliver the number of weeks late (recall, negative is early) in column 1 is 3 to 1 odds against. This is the poor bet.
- Winning the bet that the team will deliver on the weeks late in column 2 as even.
- Winning the bet that the team will deliver the number of weeks late in the third column is 3 to 1 for. This is the good bet.

For example, it is an even bet that the plan in Figure 3 will be two weeks early, a good bet it will be on time and a poor bet that it will be five weeks early.

Release Planning and Tracking

Unlike most parameter-based estimation tools, ARP is useful throughout the lifecycle of the program:

Planning

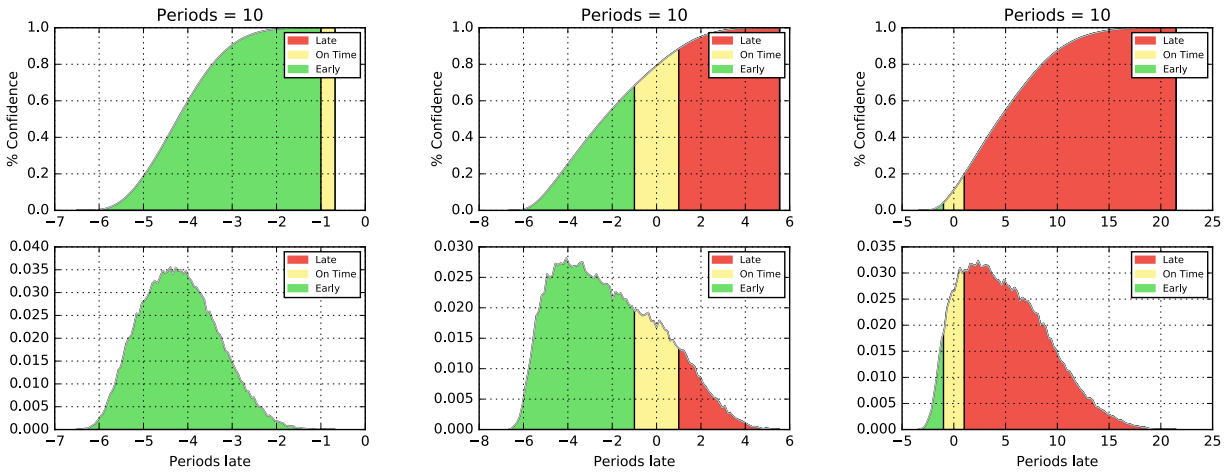
It is often important to make planning decisions about a feature without investing the time and effort for the feature decomposition. Using the Hubbard-based techniques and backlog items that have not been of ARP, one can get a good-enough estimate on the long-term prospects of delivering a set of large features and perhaps decide which ones warrant further decomposition. For example, one might find discover a desired feature is an even bet, but the safe bet is far too late. In short, there is a lot of uncertainty. In this case, you might decide to invest in the decomposition to reduce the uncertainty for making a more-informed decision.

In shorter-term release plans (i.e. 3 to 6 months), you may have to select from a backlog of decomposed features. As discussed above, in this case the decision of what features to include should be based on the business' risk appetite and the criticality of the features. Detailed plans can include partial delivery of a feature by choosing a subset of the stories.

Using Aptage for planning (Figure 7):

- If the chart is all green, the plan is too conservative. Committing to this plan essentially entails no risk.
- If the chart is all red, the plan is too aggressive with no chance of delivering on time.
- If the chart is a mixture of red and green, then there is some chance of delivering on time. The more green, the higher the odds and the safer the bet.

As shown in Figure 6, the using the output of ARP, one can easily see if the plan is too aggressive (all or mostly red), too conservative (all or mostly green), or just right (on time is a 'good bet').



	Poor Bet	Even Bet	Good Bet
Periods Late	-4.8	-4.2	-3.6

File: aptagetest10G.xlsx, Timestamp: 2016-04-15 10:33:12

Too Conservative

	Poor Bet	Even Bet	Good Bet
Periods Late	-4.0	-2.4	-0.4

File: aptagetest10.xlsx, Timestamp: 2016-04-15 10:30:21

About Right

	Poor Bet	Even Bet	Good Bet
Periods Late	1.6	4.4	7.7

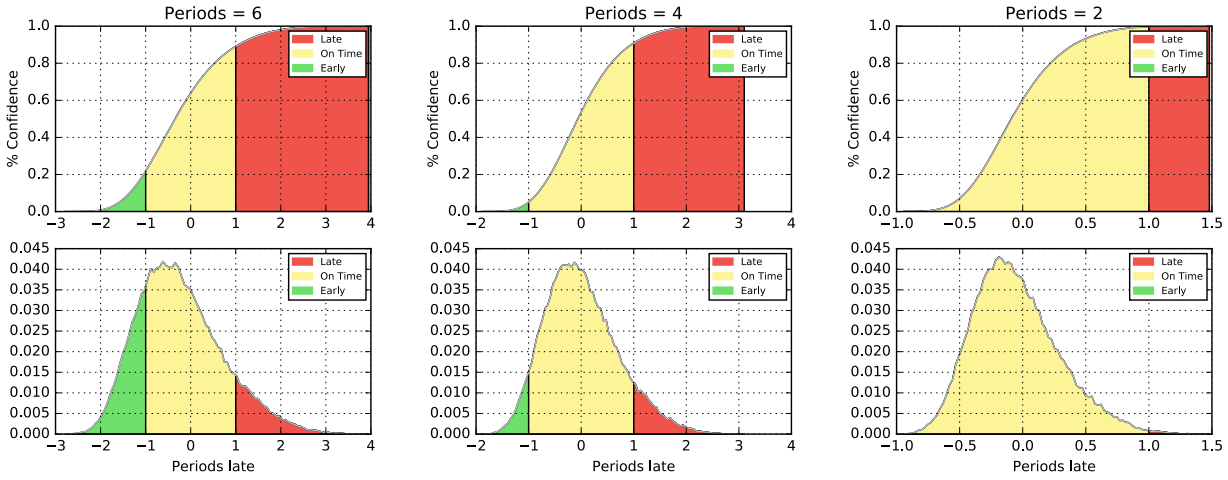
File: aptagetest10R.xlsx, Timestamp: 2016-04-15 10:31:48

Too Aggressive

Figure 7. Three ARP release plans.

Tracking

In a well-managed Agile effort, the odds of completing on-time should improve over time. Once the release is underway, ARP can and should be used to see if the bets' odds are improving for making the delivery date. An example of ARP output for a well-run project can be seen in Figure 8 below. In this example, the team decided to take on a risky, but high value project. By identifying and working off the riskier items (those with widest triplets), they were able to improve their bet by the end of sprint 2. They removed all the risk by the end of sprint 4.
These charts can be used to effectively communicate the actual status of the project to management avoiding the 'green - green - green – red' anti-pattern: This anti-pattern consists of the team taking on the less risky stories early, the low hanging fruit, to show progress. They report 'green' status at each review. When they finally get to the risky stories towards the end of the project, they suddenly have to go red. By then it is too late for the business to adjust easily. A bad time is had by all. Figure 9 shows how Aptage can give early warning of going red.



	Poor Bet	Even Bet	Good Bet
Periods Late	-0.9	-0.3	0.3

File: aptagetest6.xlsx,
Timestamp: 2016-04-15 10:39:01

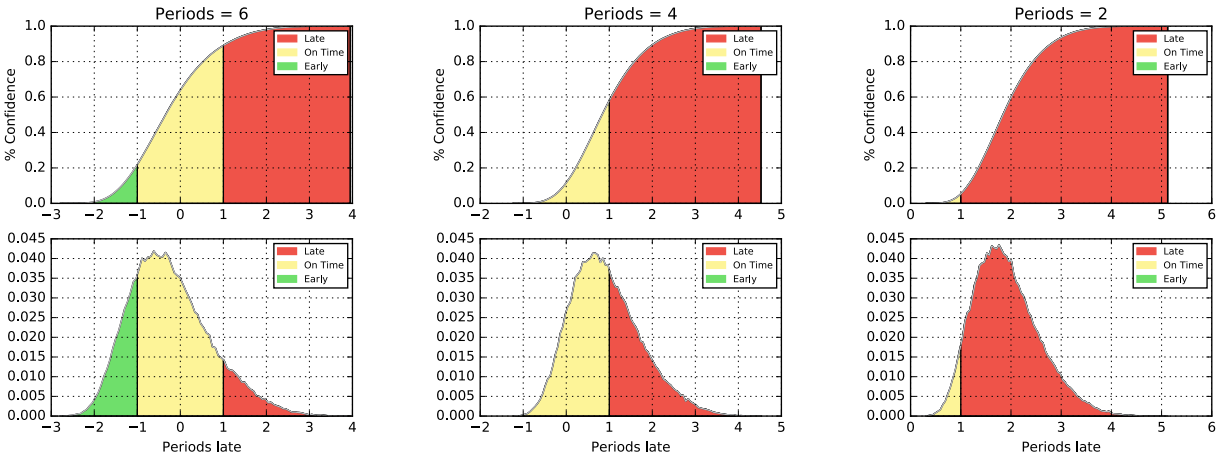
	Poor Bet	Even Bet	Good Bet
Periods Late	-0.5	-0.1	0.4

File: aptagetest4.xlsx,
Timestamp: 2016-04-15 10:41:24

	Poor Bet	Even Bet	Good Bet
Periods Late	-0.3	-0.1	0.2

File: aptagetest2.xlsx,
Timestamp: 2016-04-15 10:41:55

Figure 8: An example ARP output after several sprints of a well-managed project



	Poor Bet	Even Bet	Good Bet
Periods Late	-0.9	-0.3	0.3

File: aptagetest6.xlsx,
Timestamp: 2016-04-15 10:39:01

	Poor Bet	Even Bet	Good Bet
Periods Late	0.3	0.8	1.4

File: aptagetest4r.xlsx,
Timestamp: 2016-04-15 10:43:03

	Poor Bet	Even Bet	Good Bet
Periods Late	1.4	1.8	2.3

File: aptagetest2r.xlsx,
Timestamp: 2016-04-15 10:43:49

Figure 9. A poorly managed project

Conclusion

For development organizations to deliver valuable, innovative products, they need to take on projects that are inherently uncertain – there is not sufficient information available for precise predictions as to what can be delivered when. To embrace, rather than ignore the uncertainty, the team, their management, and the stakeholders, need a way to agree on plan that neither too conservative or too aggressive. In order to make informed decisions, the team requires a way to surface the likelihood of success for new and ongoing release plans. ARP provides that capability.