7th Agile Leadership Skills:
Set Targets and Reward Real Progress

This document is an excerpt from the book:
"Agile Project Management for Government "
Authored by Brian Werham
Published by Maitland & Strong
Reproduced with permission under license

PMCAMPUS.com / Mokanova Inc
.

Agile Leadership Behavior Seven: Set Targets and Reward Real Progress

Working software is the primary measure of progress.

Agile Manifesto Principle Seven

Targets motivate people. They need to be relevant and measurable. Most importantly, the development team needs to aim at delivering value to the stakeholders, not just the delivery of 'clean code'. Targets need to be collaboratively arrived at, unambiguous, transparent, and they should engender trust by being safe from *gaming*.

I will take these points in reverse order, and in each case compare the agile way of planning and tracking against the traditional waterfall approach.

Engendering Trust by Being Safe from Gaming

Gaming occurs when one or more parties to an agreement take advantage of loopholes to their own advantage. A team leader under pressure may log his team's work on an item as finished by prematurely handing work over to a separate testing team. A project sponsor may placate anxious stakeholders by making elegant presentations on the theoretical advantages of the end-solution, even though the initial prototypes have not worked.

The danger is that the various people on the project become *players.* Rather than working together in a collaborative manner towards an integrated goal, they start to play games with each other to gain a short-term personal advantage, rather than aligning their goals with the team's goals. One of the games played can be that of promising jam tomorrow, rather than bread today.

George Baker warns against the *distortion* of incentives, and the potential problems of attempting to trade-off risk by over-incentivizing. He refers to "The folly of

rewarding for A while hoping for B" which leads to distortion of outcomes. If measurements are being used that are not relevant to the real-world project targets, then teams often become risk-averse – sub-optimizing by focusing on short-term output targets rather than real-world delivery. In addition, no matter how many detailed measurements are tracked, it is not possible to measure all outputs. Therefore, the targets will become the focus of activity, no matter whether other measures are more important.[i]

A Stalled Project at the US Department of the Interior (DOI)

The DOI is the largest landowner in the US with 1,672 sites on 500m acres of land and 489m visitors annually. In 2003, the DOI had initiated the development of an Incident Management, Analysis, and Reporting System (IMARS) to provide a centralized incident tracking system. 6,000 staff spread across the nation had difficulty coordinating crime reporting to federal bodies, a weakness already criticized by the GAO.[ii]

The number of other bodies that needed access to IMARS information was very large. It included many internal agencies, such as the Office of Law Enforcement and Security, the National Park Service, the Bureau of Reclamation, Fish and Wildlife Service, Bureau of Indian Affairs and the Bureau of Land Management. It also involved sharing information with the FBI National Incident Based Reporting System (NIBRS), the El Paso Intelligence Center, the FBI Joint Terrorism Task Force (JTTF) Centers, the DHS Watch Office, The U.S. Court's Central Violations Bureau (CVB) and State, County, and City Governments.[iii]

The work started in 2003 with the intent of finishing development by October 2008. However, by April 2010, the project end-date had slipped by four years to December 2012.

Costs had escalated out of control. The total forecast cost of the project had increased from $23m to $42m. The project had underspent because it had not started many important activities. It often made large requests for funding but failed to manage to get the authorized work underway in time. This 'under-spending' was not money saved, but was simply the result of unattained milestones – spending was substantially below the planned cost and no progress was being made.

Things were so bad that the primary contractor voluntarily quit, and the sub-contractor was terminated for non-performance. An audit found that DOI's planning was poor, that resources were poorly allocated, and that leadership was weak.[iv]

In April 2010, the project was reviewed and put under more senior management control. The DOI worked with the OMB using TechStat sessions to review performance

and find a solution. The possibility of project cancellation was discussed.

The solution was to shift the focus from plans and theoretical milestones into delivering real solutions. Stakeholder confidence, which had been shattered by the lack of demonstrable progress, would be rebuilt by proving that progress was being made by delivering working software.

On April 29, 2010, DOI decided to pilot a working solution with 500 users by October. This accelerated deployment was successful, across five different offices in seven law enforcement programs in four time zones.[v]

The agile approach they had taken had delivered 90% of the most urgent law enforcement requirements with excellent user feedback on its usability, and several of the aging legacy databases that had been in use had been closed down.

To gain further stakeholder buy-in, and demonstrate real progress, the DOI ran a successful user evaluation exercise with five modules while operational readiness tests were completed. The department did not need to buy and install hardware and infrastructure because the three deliveries using expandable rented *cloud-computing* services that expanded as they were needed.[vi]

As this book went to press, the risk rating of the project has been reduced from *high risk* to *moderately low risk*, but challenges remain. Planning is underway for three more deliveries in 2012, and a final clean-up delivery in 2013 aimed at delivering the final 10% of *Should Have* and *Could Have* requirements. Importantly, a Post Implementation Review is planned to learn lessons from the project and to identify and recommend any outstanding actions. The real-world benefits from the resulting system, which will be used by over 6,000 members of staff across multiple Federal bodies, will be tracked in an ongoing Operational Analysis exercise. This will compare evidence of end-benefits from the use of the technical solution to the total amount of money that was spent – including the $15m and 4 years wasted in the original waterfall attempt to develop IMARS before an agile approach was successfully adopted.[vii]

An Agile Approach to Planning and Tracking

Mike Cohn stresses the importance of acknowledging uncertainty when planning. He breaks this into two types of uncertainty:

- ♦ *End uncertainty*: about **what** product features will actually be needed, and therefore which one will actually be built

- ♦ *Means uncertainty*: about **how** the product will be built. [viii]

In this way, the team avoids the mistake of imagining that features and architectures can be locked down at the beginning of a project. As time passes, Cohn points out, the end and the means become more certain, and the project concentrates on what is

actually needed.

This does not mean that we should not bother with plans, but just that we should not bother with plans that go beyond a realistic level of detail. Tom Gilb originally called this the "The principle of highest level inspection":

> "If you fail to inspect the higher levels of planning and goal-setting, then inspection at the lower levels will only serve to confirm errors made earlier! What is put into a design-or-planning process should always have exited successfully from inspection beforehand. You must be prepared to raise the clarity of planning, requirements specification, and design documentation substantially in order to exploit inspection." [ix]

Gilb proposes a formal method for identifying strategies and turning them into realistic high-level plans using the Evolutionary Project Management, or "Evo" method and the formal notation of "Planguage". Evo focuses on agile delivery with very small increments of about 2-5% of total timescale. A year-long Evo project would typically have week-long increments based on early:

♦ Identification of the highest priority requirements

♦ Ensuring that the requirements are defined well enough to be objectively verifiable. [x]

A common agile technique for measuring progress is the *burn down* chart, which simply tracks the number of story points remaining to be completed over time. Thus, the line on the graph decreases over time and extrapolates to show when the work is likely to be completed. *Release burn down* charts are for management, and show when work is released and forecast when the project will complete. *Sprint burn down charts* are team facing, and indicate the ongoing progress day-by-day during each sprint. If the line on a sprint burn down chart does not tend towards zero for the last day of the sprint (and it is unlikely to do so unless the estimation has been spot on) then the team will have to decide to change the scope of the sprint backlog.

On traditional projects, a *work breakdown structure* of the tasks and their outputs will have an estimated number of work-days and other costs against it. [xi] As work progresses the actual number of work-days and costs is recorded. Once some statistics have been built up the *earned value* can be calculated. This is the percentage of under or overspend against the original estimates, based only on work that has been completed. Partially completed tasks are excluded from the calculation, removing any optimism bias in reporting.

Agile project control turns conventional planning on its head. The features required are allocated *story points* and the team's *velocity* in delivering these story points is calculated. Thus in agile the team's efficiency is calibrated rather than the accuracy of the estimates. An alternative, then, to EVA on agile projects is a technique called *line of balance*. This technique originated at the Goodyear Tire and Rubber Company to

regulate factory processes. It ensures that the many activities in a process stay in a sustainable balance and at an even flow. This would be most useful in a large multi-scrum project. Each team's activities can be measured over time showing actual output against expected output of user stories in each sprint. This gives a graphic representation of over and under staffing across teams allowing for reallocation of staff, or more likely, a relocation of user features from one team to another.

The OMB Capital Programming Guide stresses the need for painstaking detail in the collection of enough data to calculate earned value:

> "IPTs must devote the planning time needed to create an adequate Work Breakdown Structure (WBS) at program initiation and keep it current throughout the program execution. Program management use of EVA depends on a well-developed WBS to ensure that a program is completely defined. Program experts, in collaboration with experts in the areas of cost estimating, procurement, risk management, scheduling, and EVA need to develop a WBS as a common framework within a given program, and also among related programs and across an organization's portfolio." [xii]

The Guide claims that EVA provides invaluable data:

> "For all levels of the management team … change control, performance variance, cost variance, schedule variance (etc.)" [xiii]

Alleman and Henderson proposed that this approach of measuring progress against a fixed baseline was easy to adopt for agile methods, and that they had done so in their software development company that worked in "high–ceremony government contracting environments that use EVA". They argue that as formal *artifacts* are needed for compliance with what is often a waterfall contract, EVA is a means of complying with the 'progress-to-plan' approach required.[xiv]

They see agile developments as unable to forecast future cost and schedule. They argue that this is because of the use of historical *velocity* metrics. Using this is like forecasting tomorrow's weather from yesterday's, they say. However, their argument misses the point of agile – if you fix your team size and keep to immoveable deadlines by flexing scope, then you do not need a forecast of future cost and schedule because it is fixed. EVA will tell you that you are behind schedule and overrunning on cost, even when all the functions the stakeholder need have been delivered, and on time! No schedule variance can be derived in XP because the concept has limited value in an agile environment.

The problem with EVA in the agile environment is that it is the wrong paradigm. It provides calculations of the budgeted cost for work performed. It is not *earned value* but *sunk cost*. Paul Solomon calls this the *EVMS Quality Gap*:

> "A (DOD) supplier is not required to base earned value on technical performance or quality. Technical linkage is optional in the industry standard, ANSI/EIA-748 … EV is

based on the quantity of work performed, not quality." [xv]

I have often thought that this perspective implies that "Earned Value isn't!" – isn't value and not necessarily earned that is! However, I see the value in collecting essential metrics. The velocity data collected in agile projects can do more than just be used to predicting the project completion data by use of burn-down charts. However, it is possible to use the data that most large US government projects are mandated to produce. Tamara Sulaiman enumerates the circumstances in which this might be needed:

♦ To give an estimate of progress in terms of $ burn to the team members. They will have team burn-up charts showing spend to date, and kanban charts showing feature status, but a monetary view can be an additional useful perspective

♦ If the agile project manager has to keep 'translating' agile terminology for non-Agile stakeholders used to EVA

♦ If working on a defense contract where, typically, 32 different progress reports are required to conform to the ANSI/EIA-748 standard for EVA

♦ To work out when to stop developing any further because the EVA forecasts that return from the benefits of more features is less than the spend required to develop them.

Tamara Sulaiman describes this integration as Agile Earned Value Management (AgileEVM). She claims that it is a light weight approach that compiles the standard metrics that agile projects produce as a matter of course:

♦ Current iteration number

♦ Number of story points actually completed

♦ Number of story points added to or removed from the release

♦ Actual Cost in dollars or hours. It is critical that the actual cost amount used reflects the cost needed to generate the

♦ Completed story points.[xvi]

She describes how these metrics were used by two Scrum projects. On the first project, the AgileEVM metrics were used to *calibrate* the team velocity in EVA terms. At the end of the first sprint, they proved that the scope (in story points) could not be achieved by the deadline.

The AgileEVM statistics were used in a spreadsheet model to help in requirements

trading. They focused attention on the important requirements, and identified outlying features for sacrifice. On a second project, AgileEVM data showed consistency in team output, and this gave the team confidence to calculate a final deployment date. [xvii]

Conclusions

An agile project's objectives for each iteration need to be based on providing a working solution at an acceptable delivery level. Prototypes and demonstrators have their uses early in the project, but the sponsor needs to raise expectations quickly towards piloting and operational delivery.

Various traditional project management techniques can be used to plan and track progress where they marry well with the agile approach. The AgileEVM method for analyzing earned value is an example of this. It provides traditional EVA metrics based on the data provided day by day from the agile approach.

[i] {Baker 2002 #358}
[ii] Public Law 100-690
[iii] {CIO US DOI 21/10/2005 #369}
[iv] {Faisal 2011 #368}
[v] {Faisal 2011 #368: 4}
[vi] {DOI1 2012 #441; 2-3}
[vii] {DOI 2012 #441; 3}
[viii] {Cohn 2005 #1: 254}
[ix] {Gilb 1988 #411}
[x] {Gilb 2005 #412}
[xi] Note that in the UK a *product breakdown structure* is first created to identify the outputs required.
[xii] {Ewell 2006 #159: 10}
[xiii] {Ewell 2006 #159: 50}
[xiv] {Alleman 2003 #404: 1}
[xv] {Analyzing and Measuring Information Quality #414: 1}
[xvi] {Sulaiman 2007 #403}
[xvii] {Sulaiman 2007 #403: 2}