

## Why Metrics Programs Fail

Bob Webber, Senior Fellow, Construx Software

Version 1.0, August 2012

The success of software metrics programs can vary significantly from one company to the next. There is more involved than just defining the right metrics. People at one company may embrace the metrics, track progress and use them to demonstrate real improvement. They may even propose new metrics without being prompted by management. At another company, people come up with reasons why the metrics are invalid and resist measurement, even though the metrics may be very similar. What is different? This whitepaper relates the differences to a well-established behavior model and provides recommendations for implementation of successful software metrics programs.

## Contents

Introduction.....	3
A Consequence-based Model .....	3
Applying the Principles .....	5
Summary .....	6
Contributors.....	7
About Construx.....	7

## Introduction

Metrics programs are often inserted into organizations without consideration of the human dynamics involved. The typical approach involves simultaneous introduction of a myriad of measures that increase accountability to drive results. However, the measures often serve as vehicles to punish people and teams (often in public) to get those results. People will resist measurement and comply only under pressure. Teams are unhappy and try to avoid any interaction related to metrics. Performance may improve, but it becomes a case of the proverbial “pushing a noodle up a hill.”

The author was a member of the senior R&D leadership team at AG Communication Systems, a GTE/AT&T joint venture. The division obtained national recognition for applying a human behavior model to software engineering practices. Two articles appeared in Ed Yourdon’s *American Programmer* magazine highlighting the unique approach and significant results:

“A New Approach to Software Engineering Management” - July/August 1990

“Quality Metrics at AG Communication Systems” - September 1991

This white paper will summarize the performance management principles applied to software metrics.

## A Consequence-based Model

Do your teams look forward to metrics reviews? Do they come up with their own measures? These behaviors typically are not observed in organizations. The reasons can be understood through well-established principles of human behavior. These principles can provide guidance on how to flip your metrics program into a positive experience, where people actually like being measured and produce higher results.

The scientific basis for increasing behavior through reinforcement is well-documented by Aubrey Daniels in his book, “Performance Management: Changing Behavior That Drives Organizational Effectiveness.”<sup>1</sup> It provides theory and practices that have been applied by many companies to dramatically improve results through behavior reinforcement.<sup>2</sup>

---

<sup>1</sup> *Performance Management: Changing Behavior That Drives Organizational Effectiveness*, Aubrey C. Daniels and James E. Daniels, 1992, Updated 2006

<sup>2</sup> Don’t be confused by the use of the term “Performance Management” for annual performance appraisals. Aubrey Daniels applied the term as a pioneer of applying behavior principles to organizational performance. It has since been adopted by HR departments in a different context.

In short, there are two ways to increase behavior:

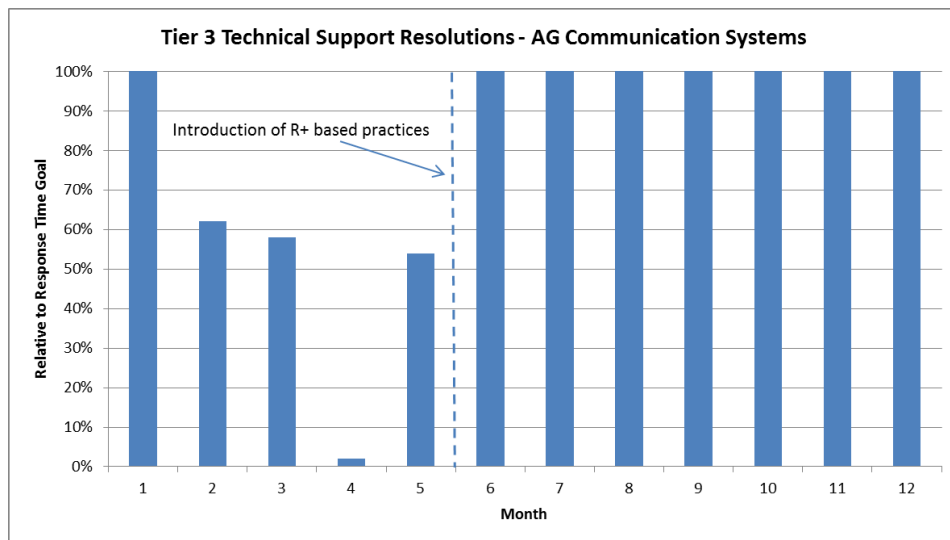
- Positive Reinforcement (R+) - People get what they want following the behavior
- Negative Reinforcement (R-) - People don't get what they don't want, meaning that they increase behavior to avoid a punishing consequence

For example, someone working towards a goal in anticipation of a reward is motivated by R+. An individual working towards a goal under fear of being yelled at is motivated by R-. Both approaches will increase behaviors to meet measurable goals, but with some key differences within the context of metrics programs. The table below contrasts metrics programs driven by R+ and R-.

R- Metrics Program	R+ Metrics Program
People resist metrics <ul style="list-style-type: none"> <li>- "This can't be measured"</li> <li>- "The measure isn't fair or complete"</li> <li>- "Too much administrative overhead"</li> </ul>	People embrace metrics. The metrics are viewed as scorecards of success and opportunities for recognition. Teams define their own metrics in anticipation of positive consequences.
People do the minimum to achieve measured goals to avoid punishment, and nothing beyond.	Generates discretionary effort. People will go above the goal, seeking higher levels of positive reinforcement.
You get results only on what you can directly measure and control. Performance is decreased in unmeasured or non-measurable areas.	People are more engaged and committed to overall success. They will find ways to measure the "un-measurable."
Continual pressure required to get metrics updated.	Metrics updated voluntarily in anticipation of positive reinforcement.
Accountability avoidance and finger-pointing.	Ownership and engagement.

**Table 1** R+ versus R- based Metrics Programs

An example of AG Communication Systems' successful application was included in Aubrey's 2006 4<sup>th</sup> edition of Performance Management. The chart below shows dramatic improvements attained in the author's department after application of performance management practices. It shows the percent of software problems resolved within response time goals for a telecommunications system of over 1M lines of code.



**Figure 1** Technical Support Resolution Example

Applying the positive reinforcement practices in this case addressed a common challenge for software development organizations. The positive reinforcement voluntarily increased attention on product support from developers who were also engaged in new development.

## Applying the Principles

In an R+ based metrics program, measurement becomes a scorecard for team success as opposed to a tool for pressure and punishment. It involves more than just casual positive recognition. It requires a well-designed set of metrics and solid practices to provide timely positive team reinforcement. What is the positive reinforcement? This can vary significantly between organizations. Here are examples:

- Baseline of current performance followed by incremental goals within reach of team to build a history of success
- Praise and recognition from management in meetings and quality forums
- Small team celebrations for achieving interim goals
- Initial small pilots with positive consequences observed by others to encourage expansion of metrics program
- Communication of process improvement success stories
- Metrics posted as scorecards for success

The last one is interesting. Have you ever wondered why metrics seem to lose effect when they are kept electronically and not displayed where people gather? The opportunities for human interaction and positive reinforcement are reduced!

What happens when either R+ or R- doesn't exist or is weak within an organization? This leads to no improvement or insignificant improvement. We often see this in organizations and call it "watching the metrics go by." One example is meetings where a long series of metrics is presented with no observable improvement and little interest - a complete waste of time and effort.

In our experience, metrics programs need to be driven by either R+ or R- to get results. Can you mix R+ and R-? You can, but it will send a confusing message to your teams, and avoidance of punishment will tend to dominate. The bottom line is whether or not your people truly believe that something good is likely to happen to them as a result of the metric. They believe based on what they experience and see happening around them - not by what you say.

Where is your organization today? If you are seeing behaviors in the R- column in Table 1, then your metrics program is dominated by perceived negative consequences.

Some typical methods of applying R-:

- Putting people on the spot in metrics review meetings when goals are missed
- Escalating missed goals to higher management
- Creating negative metrics (e.g. "Number of requirements requiring rework" versus "Percentage of requirements passing review")
- Setting unattainable goals in an attempt to get higher performance

The sad thing is that these approaches will get results if the perceived negative consequences are strong enough. If the negative consequences are reduced, performance will actually decrease, because the fear that drives the performance is decreased. This tends to re-affirm personal views that performance always decreases when harsh consequences are decreased. It's true if you are using R- as your primary motivator.

So, you can get results with an R- approach, if you are willing to accept the limitations in the first column of Table 1. An R+ approach is necessary if you want your organization to move into the second column. The R+ examples above provide some factors, but a metrics program that leverages discretionary effort should be designed from the ground up based on proven principles of organizational consequence management.

## Summary

The human behavior aspect of metrics programs is not often considered, yet it can make the difference between effective and ineffective metrics programs. Application of sound human behavior principles will make the difference between a metrics program that

yields continuous improvement and one that just becomes an administrative overhead with little value added.

How do you know whether you are in the R- or R+ column? Simple. Which of the two columns describes your organization today?

## Contributors



**Bob Webber, Senior Fellow**

[robert.webber@construx.com](mailto:robert.webber@construx.com)

+1(425) 636-0113

Mr. Webber is a Senior Fellow at Construx with over 30 years of software experience delivering innovative products in telecommunications, entertainment, and life sciences. Bob's roles have ranged from software developer to executive R&D positions at GTE and AT&T, followed by engineering and product management leadership at three successful start-up companies. Bob's record of innovation is supported by five US patents and an Academy of Motion Picture Arts and Sciences Technical Achievement Award in 1999. As CEO, Bob led TranSenda to win Microsoft's Life Sciences Innovation Award in Clinical Development in 2009, and successful acquisition in 2010.

## About Construx

Construx Software is the market leader in software development best practices training and consulting. Construx was founded in 1996 by Steve McConnell, respected author and thought leader on software development best practices. Steve's books *Code Complete*, *Rapid Development*, and other titles are some of the most accessible books on software development with more than a million copies in print in 20 languages. Steve's passion for advancing the art and science of software engineering is shared by Construx's team of seasoned consultants. Their depth of knowledge and expertise has helped hundreds of companies solve their software challenges by identifying and adopting practices that have been proven to produce high quality software—faster, and with greater predictability. For more information about Construx's support for software development best practices, contact us at [consulting@construx.com](mailto:consulting@construx.com), or call us at +1(866) 296-6300.



SOFTWARE DEVELOPMENT BEST PRACTICES

© 2012, Construx Software Builders, Inc. All rights reserved.

Construx Software Builders, Inc.

10900 NE 8th Street, Suite 1350

Bellevue, WA 98004

U.S.A.

This white paper may be reproduced and redistributed as long as it is reproduced and redistributed in its entirety, including this copyright notice.

Construx, Construx Software, and CxOne are trademarks of Construx Software Builders, Inc. in the United States, other countries, or both.

This paper is for informational purposes only. This document is provided "As Is" with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Construx Software disclaims all liability relating to use of information in this paper.