JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Book Review: The Agile Mind-Set: Making Agile Processes Work

ARTICLE

By Steve Berczuk | November 25, 2015

SUMMARY

Gil Broza's book *The Agile Mind-Set: Making Agile Processes Work* speaks to the challenges faced by people who focus on "doing agile" rather than "being agile." Rote execution of methods can only get you so far, and Broza gives insights into how you move beyond practicing agile by habit into living it.

When I speak to teams about agile software development and Scrum, I often start by discussing the Agile Manifesto. While this might seem like an extreme exercise of going "back to basics," the values expressed in the Agile Manifesto and the associated principles explain agile software development in a clear way that informs everything else about how an agile team works.

Practicing agile software development is harder than it appears, and following an agile method without a good understanding of the principles and values behind it can limit your progress. This is the central message in Gil Broza's book ***The Agile Mind-Set: Making Agile Processes Work***.

There is more to being effective at agile than following a process. As Broza points out:

Getting great at Agile, like becoming great at long-distance running or at a new career or at parenting, involves a deep shift that a sequence of small steps cannot achieve adequately. You need to make a proper commitment: “Do it like you mean it.”

He describes the agile mindset as a consistent system where values, beliefs, and principles align. This is a useful strategy regardless of your method of choice; Broza makes the point that any development approach, including waterfall or lean, is best executed in the context of a consistent mindset.

We use cookies to ensure you get the best experience on our website.

Read More

Accept

act in a way that's congruent with the values of your process, particularly when faced with challenges or conflicts on your project.

One area where organizations have a hard time transitioning to an agile process is in how they measure the effectiveness of the transition. Broza writes:

The transformation isn't a start-middle-and-end type of activity. Individuals, teams, and organizations can never be done adopting Agile, just as you can't become a responsible person once and for all.

The fact that developing and maintaining an agile approach is work that can never be declared "complete" can be frustrating to some. In practice, there are milestones and metrics you can measure to gauge how your team is doing with agile. Just as "inspect and adapt" is an approach to use with your project plan, it is also an approach that you should apply to your agile process.

But if you are using the metrics and milestones as a way to judge without an eye toward improvement, you may be missing the point. An example from the book:

A team might be following every documented Scrum practice, for instance, and still have taken on little of the mind-set. Asking "What's our velocity?" (and subsequently "Can we increase it?"). Velocity measures a team's capacity to produce output. It is not a measure of the Agility of their behavior or of the usefulness of their output. It's not even a good measure of the benefits of Agile.

Measuring velocity with the goal of increasing it in abstract closes the door to a number of possible improvements, such as changing how you define points, or correlating your velocity with customer satisfaction. Things like user experience and quality are what agile teams should be concerned about.

In addition to helping you become agile, having an agile mindset can keep you from falling back into the old, familiar ways of working that you wanted to change. It also helps you execute practices in the most effective ways. Broza points out that "in the agile mindset, being effective is more important than being efficient." Keeping this in mind can help you resist the temptation to overplan in an attempt to reduce wasted work, for example.

The key to changing how you work is realizing that agile methods form a system that helps you change how you work. Any single change can only get you so far, so teams need to make not just the changes that are agreeable at first, but also the ones that help them to change their process:

When people correctly understand that the objective is a different, whole-system approach to work, their motivation to make the change is accurate. And if that motivation is low, at least you can have an informed conversation with them and discuss the real issues.

While the book focuses on the right mindset, it also connects that mindset with tactical practices. For instance, Broza explains how timeboxing and agile planning can help effectively deliver customer value. A timeboxed sprint in Scrum forces the business to

We use cookies to ensure you get the best experience on our website.

In ***The Human Side of Agile: How to Help Your Team Deliver***, Gil Broza demonstrated his talent for explaining the intersection of process and practice. In *The Agile Mind-Set*, he helps readers bridge the gap between practice and understanding. While there is a certain value in developing agile habits, rote execution of agile methods can only get you so far. Broza gives insights into how you move beyond "agile by habit."

The message of *The Agile Mind-Set* is that you should act congruently with the principles and values of the process you are using. Broza expands on the four values of the Agile Manifesto and connects them to principles and practices. *The Agile Mind-Set* is a useful book that can help you and your team become more effective agile practitioners.

Topics: agile, agile transition, methods, process improvement,

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

UPCOMING WEBINAR

LETS HANG!

We use cookies to ensure you get the best experience on our website.

FEATURED RESOURCES

AGILE
Implementing Agile eGuide
White Paper | StickyMinds

AGILE
Leadership for Today's Agile Team eGuide
White Paper | StickyMinds

AGILE
Scrum Master Essentials eGuide
White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

2 comments

Tim Thompson | December 8, 2015 - 9:53am EST
Maybe the book gets to practical applications, but measuring means always compare an unknown to a known. So how does velocity measure when most of the work done is brand new development? What is the point there to establish a velocity measure that in the end means nothing because the team will not do similar work ever again? Velocity and estimates are great when all we do is CRUD pages with a set amount of fields. Most applications are not that simple minded. Even if a team knows **a velocity** (means not **the** velocity) will that make stories get completed faster? A unit of work will take as long as it does to be completed no matter what the velocity or the estimate is.

From my experience focusing on what the business needs is key here. If the business needs feature A and putting that into place takes five weeks then take the five weeks rather than spend endless amounts of time on trying to shoehorn this into an artifical time box. Especially SCRUM adds so much overhead and process that it typically

We use cookies to ensure you get the best experience on our website.

parallel and put the focus on getting stuff done regardless if it takes a day or a month. The biggest benefit of Kanban is that it allows for crafting stories that generate value once they are done. Sure, the team could build the database schema in three weeks and it can be delivered, but without a frontend it generates zero value to the user.

Focus on business needs, delivering value, and limiting process to the minimum.

Log In or Register to post comments

Steve Berczuk | December 9, 2015 - 1:53pm EST

A couple of points. Point estimates are meant to be based on similar sized stories (units of effort, etc), and while any relative sizing is subjective, some teams do a good job of estimating consistently. Another approach is No Estimates, where you don't estimate. But you still need to put some effort into sizing the work, so your measure of velocity is more like 'stories over time.' I don't agree that Scrum is less agile than Kanban. They address different problems. Many teams have problems with getting to incremental deliverables, and thus incremental feedback. Having said all that, the points the book make are valid for both Scrum practitioners and Kanban practitioners. And in spirit for non-agile method users as well.

Log In or Register to post comments

YOU MAY ALSO LIKE

AGILE

From Silence to Self-Organization: What Social Interaction Games Achieve That AI Cannot

Dinesh Sharma | June 17, 2026

AGILE

Why Your Agile Team Is Failing Even When Everyone Knows Their Stuff

Ella Mitkin | March 2, 2026

AGILE

Solving the Problem vs Treating the Symptoms

Payson Hall | January 16, 2026

We use cookies to ensure you get the best experience on our website.

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.