



Supplementary resource for the book *The Agile Mind-Set*. More at [www.TheAgileMindsetBook.info](http://www.TheAgileMindsetBook.info)

## How Technically Agile Is Your Team?

### Self-Assessment Questionnaire

To enable business and planning Agility, certain elements of team-level technical execution are needed. This brief assessment will help you see how technically Agile your software team is, with respect to their product, and how critical it is to pay attention to that dimension now. You may want to conduct it with your team (perhaps averaging their independently given responses), to gain as much objectivity as possible.

#### **How soon before the team can determine that a build can be safely deployed to production?**

(4 = Less than a day; 3 = Less than two days; 2 = Less than one week; 1 = More than one week)

#### **How frequently do team members improve the code's design?**

(4 = As part of most tasks; 3 = Several times per iteration; 2 = Every few weeks, they take on a refactoring task; 1 = Hardly ever)

#### **How confident do team members feel about refactoring or modifying existing working code?**

(4 = Highly confident; 3 = Somewhat confident in most areas; 2 = Largely hesitant; 1 = Not at all; they avoid it)

#### **How suitable is the architecture to the product's actual needs and evolutionary path?**

(4 = It's just right; 3 = It's over- or under-engineered, but generally suitable; 2 = The two slowly diverge; 1 = Nothing is easy to do or change)

#### **If the integration between two important components is changed, how quickly can the team verify that the components interact correctly?**

(4 = Less than 30 minutes; 3 = Less than half a day; 2 = More than half a day; 1 = More than 2 days)

#### **How high is technical debt?**

(4 = Nothing significant; 3 = Some of the older areas are hard to adapt, but newer parts are in decent shape; 2 = It's hard to make progress; 1 = Almost everything we do has repercussions)

#### **How is knowledge spread around the team?**

(4 = We're all specializing generalists, and our needs are covered; 3 = Most people are generalists, but several components are siloed [owned by individuals]; 2 = Only one or two people can contribute in all areas; 1 = Everyone owns certain components and can only work there)

Visit us at: [www.TheAgileMindsetBook.info](http://www.TheAgileMindsetBook.info) and [www.3PVantage.com](http://www.3PVantage.com)

© 2013 Gil Broza

Email: [gbroza@3PVantage.com](mailto:gbroza@3PVantage.com)

**3PVantage**

# The **AGILE MIND-SET**

GIL BROZA

*Making Agile Processes Work*



Supplementary resource for the book *The Agile Mind-Set*. More at [www.TheAgileMindsetBook.info](http://www.TheAgileMindsetBook.info)

## **How challenging is product complexity?**

(4 = Most team members can follow most logical paths without too much effort; 3 = There are many logical combinations, yet we have most paths under control; 2 = We must rely on documentation and testing to figure out many areas; 1 = Oh, dear!)

## **When the team contemplates further development in the next few months, what's the dominant emotion?**

(4 = Bring it on! 3 = There's going to be work, and some areas will be challenging; 2 = We're not looking forward to it; 1 = High anxiety)

## **In your recent product release, development iterations (sprints) were probably followed by a "hardening" period for stabilization and release-readiness. Divide the length of the hardening period by the total time spent in development iterations. What do you get?**

(4 = 0-5%; 3 = 6-15%; 2 = 16-49%; 1 = 50% or higher)

## **Your Score**

Sum up your numbers and divide by 10. Where do you stand with respect to the following benchmarks?

- Extremely Agile teams working on rapid-cycle platforms (e.g., Web) on relatively new products will score between 3.5 and 4.
- The average technically-Agile team would score around 3. Their use of Agile, while suboptimal, works well for them and will not degrade much over time.
- A score between 2 and 3 means that your team's technical execution doesn't allow for effective Agile development. Cycle times will increase gradually; expect to look at rewrites in a couple of years.
- You can expect Agility to stall over the next year or two if your score is 2 or less. The closer it is to 1, the higher the likelihood that Agile is the wrong paradigm for your team with their product.

## **Comparison**

Now take a few minutes to recall the situation exactly a year ago, then score the team and their product again as if done at that time. Are you seeing a substantial improvement? Or a bounce-back?

## **Next Steps**

If you find that your team's technical Agility gets in the way of either planning or business Agility, commit to increasing it. Set a 6- or 12-month goal for improvement, either in the overall score or in the answers to particular questions above. Increasing technical Agility requires considerable investment and dedication, so start with small steps. Invite Gil Broza ([gbroza@3PVantage.com](mailto:gbroza@3PVantage.com)) for a free strategy session to help you draw a practical roadmap. Also, check out his unique training at [www.AgileEngineeringCourse.info](http://www.AgileEngineeringCourse.info).

Visit us at: [www.TheAgileMindsetBook.info](http://www.TheAgileMindsetBook.info) and [www.3PVantage.com](http://www.3PVantage.com)

© 2013 Gil Broza

Email: [gbroza@3PVantage.com](mailto:gbroza@3PVantage.com)

**3PVantage**