Test-Driven Management

by Joshua Kerievsky & III
Industrial Logic, Inc.
November, 2002

Note: This article will be published by Software Development Magazine (www.sdmagazine.com) in early 2003.

Extreme Programming (XP) teams gain direction and confidence by being test-driven. Teams learn what needs to be built by writing tests and gain confidence in their work by running those tests. This has worked so well in practice for customers and programmers on XP projects that it begs the question "Can managers also benefit from being test-driven?"

Like much of what makes up XP, test-driven management lends a new name to an old and sound practice. Test-driven management is a practice for development managers to help ensure that the software their teams produce meets the organizational intentions of business managers (also known as Gold Owners). Why would XP require this?

Test-driven management is necessary because Gold Owners frequently don't communicate their organizational intentions and business objectives to XP teams. While Release Planning does an excellent job of elucidating *what* features are needed, it falls short when it comes to explaining *why*. For example, scoring a number-one ranking from a premier industry reviewer or achieving 40 percent market share by second quarter next year are clear business objectives—yet nothing in the XP process keeps such objectives on an XP team's radar screen. As a result, business objectives that ought to inform a team's decisions and actions fail to do so because they go unarticulated. When that happens, the team can produce finely crafted, fully tested software that doesn't deliver on financial or organizational expectations.

Test-driven management addresses this problem by turning the knobs up on two of the XP values: Communication and Feedback. Managers on XP projects use test-driven management to clearly articulate and assess organizational intentions and business objectives for their projects and teams. Not only does this give them and their teams' confidence that their projects are preceding according to expectations, it also forms vital connective tissue between:

—Management (including executive, business and development managers),

—Customers (including subject matter experts, marketers and salespeople, support staff and analysts, testers and quality assessors) and

—Programmers (including programmers, testers, administrators and database administrators).

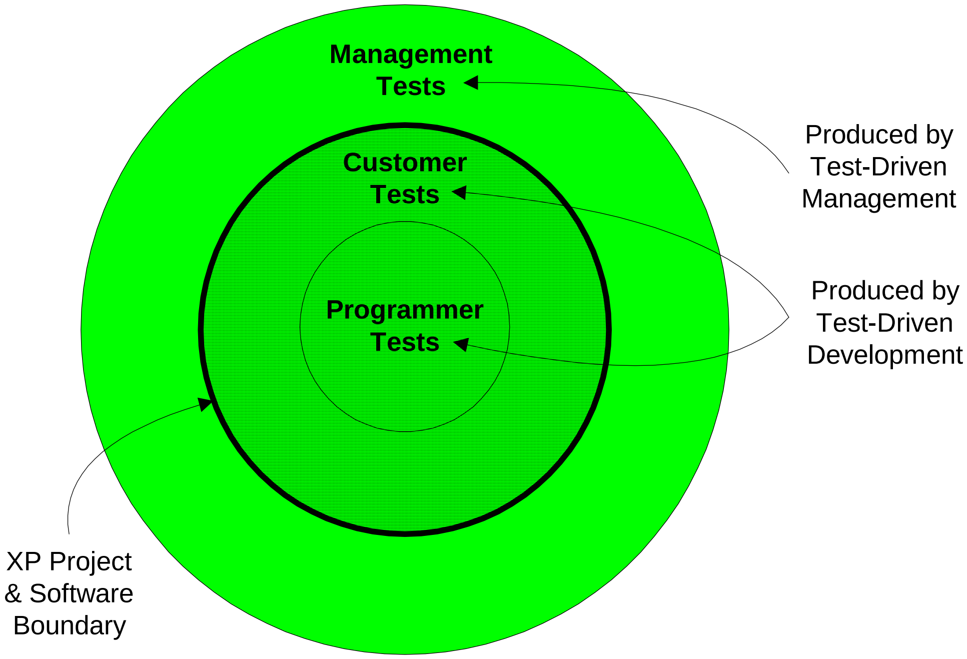
Test-driven management directs the specification of management tests. A management test is a statement that indicates a measurable, time-limited goal, framed in a binary manner: We either achieve the management test or we fail. Good management tests are SMART: Specific, Measurable, Achievable, Relevant and Time-Based.

Management tests are statements about the world external to the project, treating the project as a boundary of responsibility and authority. They avoid specifying ways in which external effects (that is, things that occurs outside the boundary of the project and the software) should be achieved. In other words, good management tests set a destination, but don't specify how to get there.

Management tests are specified using spoken language, not software, and must be measurable and undebatable. The measure can be a hard number (50 customer organizations will have signed up to use our software by December 31st) or a perception number (by project end, survey of customers' happiness shows a 20 percent increase on a scale of 1 to 10).

Management tests provide an excellent way for the entire XP team—managers, customers and programmers—to understand what unites them. This echoes Tom DeMarco and Timothy Lister's observation that "The purpose of a team is not goal attainment, but goal alignment." (*Peopleware*, Dorset House, 1999, second ed.). Management tests create goal alignment by delineating how and when success will be measured, enabling individuals to understand the effects of their own actions.

Management tests complement an XP project's customer and programmer tests by adding a test layer around the project itself. While programmer tests assert that small units of code meet programmer expectations, and customer tests assert that whole system features meet customer expectations, management tests assert that organizational returns on investment meet management expectations:


Management Tests
Customer Tests
Programmer Tests
Produced by Test-Driven Management
Produced by Test-Driven Development
XP Project & Software Boundary

With test-driven management, XP teams elicit, clarify, negotiate and codify management tests from their business managers before a project begins. They “elicit” these tests rather than create them, because they are ultimately answerable to business managers, who provide project funding and resources. What does a good management test do? Ideally, it will balance what the organization would like to accomplish on any given project against available resources. Such balancing work is properly captured in a project charter (more about this in a later article).

Organizational accomplishment comes in two flavors: external and internal. External management tests are focused on the success of the host organization, measurable outside the project and software boundary, and critical to the organization’s Gold Owners.

Internal management tests are focused on the success of the development organization, measured at or within the project boundary, and critical only to development managers.

We recently had a chance to help an XP team formulate a suite of management tests for the company’s first XP project. Many of these tests were written to answer a thoughtful and important question posed by a Gold Owner: How will we know that XP is better than our current process when the pilot project ends? Here are a few of the tests we wrote:

Customer Participation
By the start date of project, 10 companies sign up to be active external customers to the project. (External Management Test)

Team Agility and Productivity
External customers report that the XP team is more responsive and productive by an increase of 25 percent at the end of project compared to initial survey results. (External Management Test)

Team Satisfaction with XP
At the end of the project, 70 percent of project participants report eagerness to do XP again. (Internal Management Test)

Quality at Release
Released features meet a “mean time to failure rate” of not more than 0.1 major errors per hour, as determined by testers outside the team.
There are zero category-one errors remaining in the code at release, as reported by testers outside the team; all errors of other categories number 20 or less.
(External Management Tests)

Knowledge Transfer and Learning
A survey of participants before and after the project clearly indicates that team members’ learning and proficiency has increased two points on a scale of one to 10 in different areas of code by project’s end.
(Internal Management Test)

These example management tests were originally more general, and were honed to precision by continual refinement over several days. For example, here’s an initial attempt at an internal management test of the pilot team’s open workspace:

Open Workspace Creation
By the project’s start date, the team’s open workspace has been established.

This test suffered from two serious flaws:

- It’s a development management *deliverable* rather than a measurable test.
- It needs some empirical statement of measurement.

Here’s what we came up with after revising the test:

Open Workspace Acceptance
By end of project, 90 percent of team’s participants report that open workspace increases their personal productivity as compared to working in cubicles.

The test now clearly states a measurable effect of the project.

To help foster alignment among the pilot project's participants, all of the management tests were placed on large, visible posters in the XP team's open workspace. At various times throughout the project, when management wanted to know how things were progressing, the management tests made that easy: To report project status, we simply assessed the management tests and reported results. Business managers and executives, who hadn't been involved in the writing of the management tests, were thoroughly satisfied with the tests, which provided a dashboard to measure the project's progress.

At one point in the project, the team wasn't doing very well on the following external management test:

Active Customer Feedback
Per iteration, 75 percent of participating external customers voluntarily provide commentary on the system.

About one month into the three-month-long project, we were getting feedback from only about 30 percent of our 11 external customers. That wasn't good, for a team can't be very responsive to customer requests (another external management test) if it isn't getting active feedback from its customers. We looked into the problem and found that a good number of the external customers weren't being shipped CDs of the software (at each iteration end) because of a licensing issue. Resolving that matter was no trivial task; so, even at project end, we didn't succeed at the Active Customer Feedback test.

Not passing a management test provides valuable information to the whole team. Of course, it's good to know this information during rather than at the end of a project, while an opportunity to correct the problem(s) is still viable. On the pilot project, the coach and development managers assessed the management tests at the end of each week, since iterations were one week long. At the end of the project, during a retrospective, the whole team reviewed the management test assessments, learning much valuable information.

We also learned another lesson. The way the team chose to pursue one of its internal management tests contributed to the failure of an external management test. The successful internal test involved the team's open workspace (see Open Workspace Acceptance, above). The failing external test was the following:

Buzz Factor
At least 50 percent of surveyed non-pilot employees express interest in engaging in future XP activity by end of pilot project.

In pursuit of the Open Workspace Acceptance management test, development managers needed to create the team's open workspace. The trouble was, space was at a premium on the second floor where the project's participants currently worked in cubicles, near their colleagues. So, the coach and development managers decided to establish the open workspace on the first floor, where space was available. The pilot team moved down to the first floor, leaving their colleagues behind.

At the time, the team didn't realize that this move would contribute to the failure of the Buzz Factor management test. Placing the open workspace on the first floor, away from the non-pilot employees, reduced the buzz factor to a whisper, since many of the employees didn't take the time to come down to the first floor and check out what was going on with the pilot team. The failure of the Buzz Factor test, which was exacerbated by an inappropriate joke from a business manager, led management to make an important decision at project‘s end: Relocate the open workspace to the second floor. This decision arose in planning for new management tests for a phase II of XP's expansion in the company, after management evaluated all of its management tests and learned that the pilot project had been an overall success.

The XP pilot team also wrote tests to focus on optimal XP performance. Consider the following internal management test:

Productivity Consistency
For the final three iterations, actual team velocity is greater than or equal to estimated team velocity.

Passing this test involved getting good at estimating work during XP's Iteration Planning Game. The test pinpointed the final three iterations of the project in light of the fact that few teams can accurately measure their velocity during early iterations. Once the team could accurately estimate its velocity from iteration to iteration, it could demonstrate productivity consistency. Development managers value such consistency because it allows them to do more accurate Release Planning.

Management tests should never state what system features must be completed by a given date, for that's the job of Release Planning. The tests should also not create an environment in which people live in fear of the tests. Management tests are meant to align the intentions and actions of a whole team, which means that the team must be comfortable in their willingness to commit to the achievement of the tests. It's therefore best for XP teams to formalize management tests with their business managers prior to embarking upon Release Planning.

Test-driven management addresses a shortcoming of XP: accountability to business needs and realities. The best way to resolve this problem is to invite business managers to specify tests that guide development teams and certify their work's ultimate business success.

Test-driven management and test-driven development work hand in hand to provide direction and confidence to entire project communities. By complementing your customer and programmer tests with management tests, you'll have a better chance of achieving important organizational objectives.

Authors:
Joshua Kerievsky (joshua@industriallogic.com) is a software development coach and programmer. After programming on Wall Street for nearly 10 years, in 1995 he founded San Francisco Bay Area–based Industrial Logic (http://industriallogic.com), a company that specializes in Extreme Programming. Kerievsky speaks regularly about XP, has authored numerous XP articles, simulations and games, and authored the forthcoming book, *Refactoring to Patterns*.

III is Senior Jiggler at San Francisco–based Systemodels, liberating people and communications in the systems environment, and working on presentations of system ideas that ignore technology and focus on clarity. He can be reached at (415) TANTRUM.

Acknowledgements
The authors would like to thank Kent Beck and Ward Cunningham for reviewing this article before it went to press.