



May 19, 2004
3:00 PM

“TEST-DRIVEN PROJECT MANAGEMENT”

**Scott Lazenby
Glenayre Technologies, Inc.**

Scott Lazenby

Scott Lazenby is the Test Engineering Manager for Emerging Products at Glenayre Technologies, Inc. In addition to his test management duties, he is also heavily involved with the Quality department's process improvement program and is responsible for engineering tool implementation at Glenayre. Prior engineering tool experience includes: Mercury WinRunner and LoadRunner implementation for Glenayre web-based products; SS7 call-processing tool implementation for the telecom-based products; Creating a linked summary report between the test results database and the defect tracking database; and integrating a customer-service ticket escalation mechanism between the customer-service CRM system and the engineering department's defect-tracking tool. Scott is currently evaluating model-based testing tool options.

Scott attended the University of Georgia where he received his BA Management Information Systems in 1994. Scott's first visit to StarEast was in 2003.

Track: TESTING IN THE AGILE WORLD

Title: Test-Driven Project Management

While the test organization is normally considered the “Subject Matter Expert” within a software company, it is rarely charged with leading a software development effort. In fact, with the increased popularity of “Extreme Programming” and specifically the concept of “TDD: Test-Driven Development”, many testers are working to expand their skill sets so that they can adapt to a changing test culture where they will be viewed as part of the Development organization.

Test-Driven Development is a development technique with which you must first write a test that fails before you write new functional code with a side effect of ensuring that your source code is thoroughly unit tested. A funny thing happened at our company this year when the Development organization presented to Senior Management how they wanted to implement “Test-Driven Development”. In their version, the test team would begin reporting to a development manager who would assign them unit-testing tasks. Priority for assignments would be made for the unit-testing effort, as the development manager was granted final approval. Needless to say, this process never got approved and Senior Management came up with an alternate approach. Rather than limiting “Test-Driven Development” to unit testing driven by development, senior management felt that it should ask the test organization to explore product launch-related issues such as packaging, system configuration, upgrade procedures and documentation at the beginning of the project. As we began work on our product platform migration over to Linux, senior management continued to emphasize their expanded view of “TDD” and as a test organization we embraced this view as “Test-Driven Project Management”. With “Test-Driven Project Management”, the test organization can lead the project management effort by serving as a catalyst for involving customer service earlier in the process; resolving issues between architecture, development and product management such as requirements validation and coverage; and winning over the confidence of senior management so that proper expectations can be set and met. This presentation shares the experiences we have encountered thus far with implementing a “Test-Driven Project Management” environment and how we now have the confidence of the CEO to the point that he frequently checks in with the team on progress and values our view of the complete product.

Four keys to implementing a Test-Driven Project Management process include:

- Team Building
- Establishing credibility with Senior Management
- Risk Management and Reporting
- Rallying the Test Organization

Team-Building

To successfully drive any project, you must first form a strong core team. Most core teams consist of only department leads or managers. Most communication between departments is unfortunately limited to the interactions of these core team members. For our test-driven approach, special care was taken in building a multi-faceted test team that was not limited to one team lead or skill set. That way proper attention could be paid to each department. Roles created in our test organization include:

- **Project Leaders:** Duties include representing test team at core teams, defect-analysis and any other inter-department communication meetings. They are also expected to proactively seek feedback from customer service, development and product management.
- **Designers/Technical Leaders:** Responsible for laying out the test architecture for the projects. They outline the scope of the work to be done and spend the majority of their time with the architecture and development teams. Focus on testing metrics such as requirements coverage, troubleshooting issues with development and automating testing tasks.
- **Functionality Testers:** Interact often with both development and customer service. When finding showstopper issues they make sure the issue is thoroughly diagnosed and the proper workaround information is distributed throughout the organization. They also try to find ways to assist development in resolving issues faster. To assist in training other departments such as customer service, technical communications and training, this group will often invite team members over to assist in executing tests. By executing tests together, installation and troubleshooting documentation can be updated in a more timely and accurate manner.
- **Performance Testers:** Focus primarily on load testing and reliability. In our organization, it requires a development background to create and maintain test automation scripts, reporting tools and code-analysis data. Members of both the test and development organization make up this team.

From the onset of the project, the test team was asked to drive project issues even if it meant picking up another department's normal duties. This created some tension between some test team members but as progress was made and issues were resolved, that tension was replaced with the satisfaction of knowing that we did not encounter the same problems we had previously. Prepare the test team for their new responsibilities by assisting them with setting up the initial inter-department communication meetings and share with them the excitement and gratitude given by the other department leaders as you pitch the test-driven project management concept to them. Some ways we create rapport with other departments include: inviting them to your test planning meetings, sharing all project-related information with them as if they are on your team, and volunteering to help when they are behind schedule.

When offering suggestions, make a point to follow-up with action from your group. For example, after rejecting a fix for a GUI screen defect for the second time, one of our test leads who had a background in web design, offered to update the GUI screen, clear it with Marketing and save the already overworked developer rework time.

Finally, admit when you are wrong. The truth always surfaces in the end and the goal here is to reduce stress so why prolong it? Remember change will not occur overnight and you might not win over every department the first time out.

Establishing Credibility with Senior Management

In order to gain the confidence of Senior Management, you will need to first gain exposure and secondly offer information that will help them to succeed. If you are a new manager or are under new management, make sure to introduce yourself to all other department heads. It is amazing how scared most employees are of senior managers. The simple act of common courtesy may actually launch your own career faster than you ever imagined. The goal here is to make a lasting impression, so make sure any interaction is productive. As part of introducing yourself as the test manager, offer up how proactive you like your organization to be and maybe share a short example. Keep the initial meeting positive. Test groups tend to be pessimistic by nature so avoid this tendency, as it will leave a negative impression. Another good idea is to offer to send a link to your weekly report or maybe the monthly high-level project review presentation. The sending of the link will remind the senior manager of the initial meeting and provide something you can discuss at a future meeting.

After gaining your initial exposure with senior management, be prepared to see them hanging around your group more frequently. You will need to be able to share a quick status that will provide them with confidence that the project is making progress and avoid causing any unnecessary panic by sharing too much information related to a current issue that the team is dealing with. Pay attention to your body language and tone and keep the interaction brief. Always have a high-level status report rehearsed for the day and make sure your team is coached appropriately on interactions with other department managers and your own upper management. Establish a communication policy for your team to follow or depending on the makeup of your team, you may ask your team to route any requests directly to you.

If you are part of the senior staff project review meetings, make sure you are always prepared. Maintain a high-level status of all projects and meet frequently and informally with your team members. Do not feel the need to attend every status or task-level meeting. Trust your leads to provide you with the correct details. While the high-level status report is the focus of a senior staff meeting be prepared for tough questions and bring along supporting data to back your report. You will be challenged on any status presented. Sample high-level and low-level reports are covered in more detail later in this paper. Being prepared will also improve your confidence going into the meeting. Another useful activity is to meet with your peers from other departments prior to senior-level meetings to ensure communication is consistent. Notify others if there are any red alerts in their status reports that may trigger a grill session.

During the senior level meetings, focus on the big picture. Avoid discussions on personnel issues, playing the blame game or trying to solve an identified problem in the meeting. Report on how the project is tracking to schedule and if you are off schedule how will the team get it back on track.

Showing respect is another easy trait that too many people take for granted when trying to gain credibility. While most new managers do the following things consistently, some veteran managers can become complacent and find themselves actually losing credibility.

- **Always arrive early to meetings.** Not only does it show that you are prepared, it can also allow you some quality chat time with the same senior managers you are trying to persuade.

- **Mind your manners.** In the software business, deadlines and pressure is a given and within your peer groups, informality is accepted and expected. However, do not maintain that same persona when engaging with senior staff.
- **Listen actively.** While most successful managers do tend to carry on a little too long when emphasizing a point, they are also savvy enough to know when someone is not paying attention to them. Do not be the person who stammers to answer a question about the topic just discussed.
- **Follow-up.** Follow through on any action items assigned to you. You will quickly lose credibility by not completing or being late on an action item. If the action item involves a corrective action, make sure the action is legitimate and if necessary request additional time to resolve the issue.

Risk Management

By taking on a project management role, the ability to manage risks will determine how successful your project will be. At the beginning of a project, it is easy to meet with other departments and come up with the usual laundry list of project risks. Even with a risk mitigation plan, some issues will inevitably occur and they will need to be reported, as they will impact the project schedule. In representing the test team, whose job is to validate the readiness of a product, you will most likely end up being the one who will have to raise awareness on project risks before they spiral out of control. To avoid alienating your team when escalating issues you need to first establish an escalation policy with the other departments. The policy should clearly state that escalations are intended to raise awareness on a company problem and not be used to blame others. Avoid the extremes of being too lenient or too harsh when discussing the need to escalate issues as they occur. Be prepared to adjust the policy depending on the personalities that you are dealing with and remember it is always better to err on the sensitive side. Let them know up front that any risk that later becomes an issue will need to be reported as early as possible. It is better to digest bad news earlier knowing that you will have more time to resolve the issue.

Prior to escalating any issue, make sure you notify the department responsible for the issue and offer to help resolve it. If you can point to a proposed solution when escalating an issue, you can avoid the appearance of attacking or discrediting the offending department. The goal is to attack the problem and not the person.

Reporting

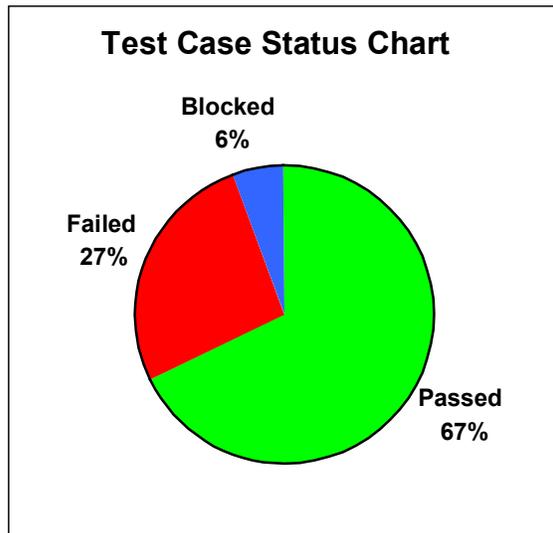
As mentioned earlier in establishing credibility, the ability to report project status to a variety of audiences is an effective way to win over other departments. Most senior managers believe they have some technical proficiency and by helping them understand the project issues better, they will be more likely to help you resolve them. Limit volume of data presented to a High Level View but have the detailed reports available. You never know when you might encounter a former engineer who wants to show that he can still stay on top of the technical details. Seeking feedback about how your audience is expecting data to be presented will also save you time.

As a test manager, you should already provide trend data such as the incoming defect rate, the defect resolution rate and test summary data. When raising issues, focus on the business impact. Will this failed feature impact a pending order? Ensure your data is accurate and be prepared to defend it with facts. Another proactive technique is to publish your results frequently and preferably online. Note that should you choose the online option, you will need to make sure any data published is current with the latest data.

Some detailed reports may also help development to prioritize work. For example, while testing our voicemail application we found that a good way to divide the workload was to organize the test cases by menu item and then subdivide them by function. That way within a menu we could paint an accurate picture of how many functions in a given menu were valid. We then incorporated the menu name as part of our naming convention for the headline field when writing defect reports so that we could quickly match defect fixes to test cases. In our example, all defect titles would start with the top-level menu so that we could simply sort by headline. Customer service was asked to gather statistics on menu usage so that menu priorities could be established. The resulting report then allowed development to knock out multiple defects at once and made sure that the most critical menus were addressed first. Pages 7-10 contain sample reports starting with a High Level view and then working down to a detailed view.

High Level Test Summary Report

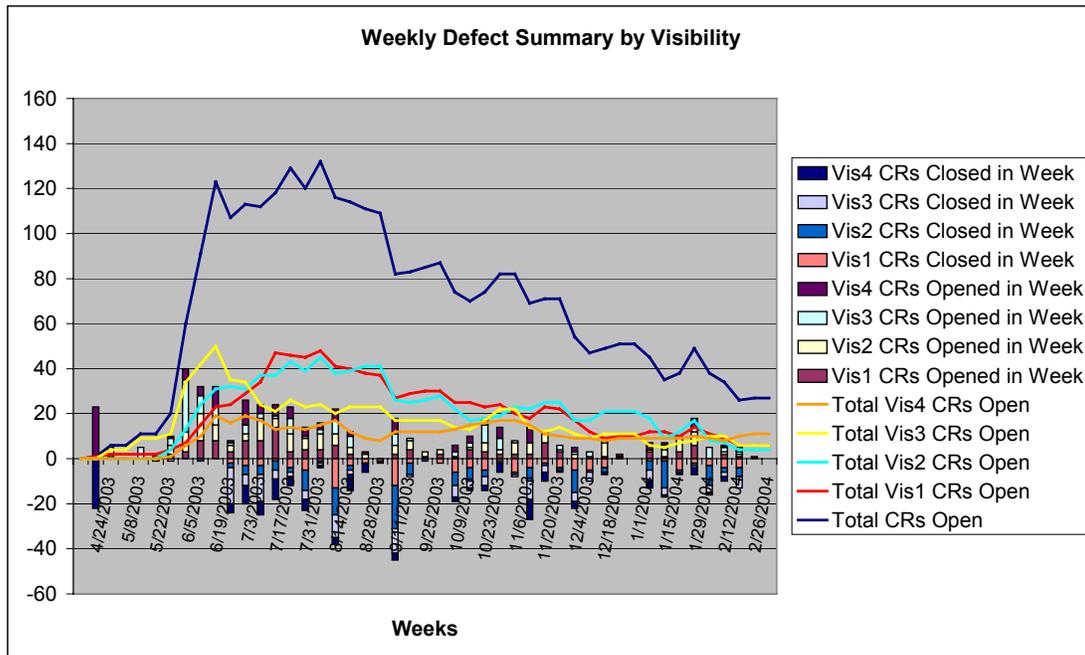
The sample here is from a monthly PowerPoint presentation that is given to Upper management. The Test Case chart shows the overall state of all test cases for a given application. We generally target a minimum 85% Pass rate before agreeing to launch a beta site. The data table on the right shows how successful the most recent code submissions were in terms of defects submitted, fixes approved, fixes rejected and number of new defects found. We also include bullet point summaries of any outstanding issues that may impact project delivery with this report.



Defect State	Build 1	Build 2
Submitted	40	39
Pass Test	-37	-30
Reject Test	3	1
In Test	0	8
New defects	48	26
Defect change	14	-3
Reject Rate	8%	3%
New Critical defects	15	3
Submitted Critical defects	15	5

High Level Defect Trend Report

This is a standard weekly defect trend chart that shows total defects opened and closed. Visibility is how we prioritize issues with Visibility 1 issues ranking as the top priority.



Detailed Defect Breakdown Chart

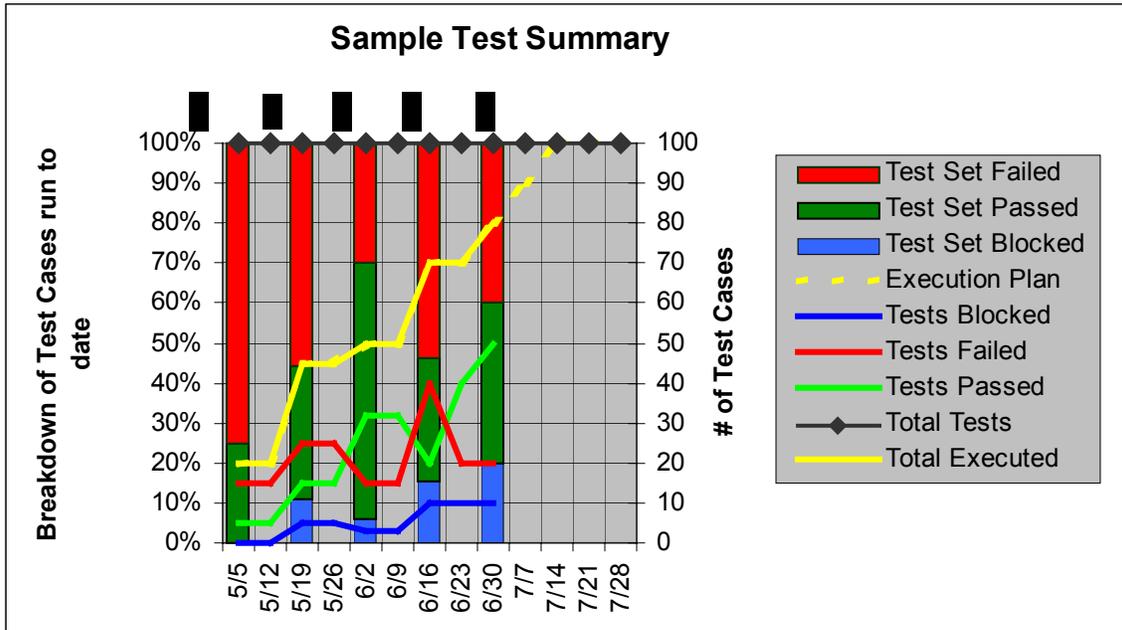
Surprisingly, for such a detailed data table of the open defects and their respective states and priority, Senior Management often requests this chart to correlate with the Weekly Defect Trend Chart. Perhaps it is because it also includes a work queue field that shows which team is responsible for the next state change.

Work Queue	State	Unranked	V1	V2	V3	V4	Total
Defect Review Team	To_Be_Reviewed	23	0	0	0	0	23
Development	Reviewed:	0	11	60	63	11	145
	Confirmed:	0	4	7	6	37	54
	Assigned:	0	6	11	40	16	73
	Opened_In_Engineering:	0	20	29	16	17	82
	Data_Pending:	0	9	11	8	2	30
	Rejected_By_Test:	0	2	5	1	2	10
	<i>Development Work Queue:</i>	0	52	123	134	85	394
Test	Submitted:	0	48	39	14	1	102
	Opened_In_Testing:	0	13	9	2	1	25
	<i>EPA Work Queue:</i>	0	61	48	16	2	127
Quality	Fixed:	1	14	12	14	2	43
	Pass_Test:	0	2	0	1	0	3
	Rejected_By_Engineering:	0	3	6	19	3	31
	Duplicate	0	12	7	8	0	27
	Completed:	0	1	0	0	0	1
	<i>Quality Work Queue:</i>	1	32	25	42	5	105
Total CRs:		24	145	196	192	92	649

Detailed Test Case Summary Report

This is our "Top of the Mountain" chart. The trend lines show total number and percentage of: total tests to run, how many have already been executed, an estimate for when the remaining cases will be run (Execution plan), and the number of the total tests that have passed, failed or are blocked on a weekly basis.

The bar lines show the percentage of test cases passed, failed and blocked per build which is labeled on the top axis.



	5/5	5/12	5/19	5/26	6/2	6/9	6/16	6/23	6/30	7/7	7/14	7/21	7/28
Total Tests	100	100	100	100	100	100	100	100	100	100	100	100	100
Tests Passed	5	5	15	15	32	32	20	40	50				
Tests Failed	15	15	25	25	15	15	40	20	20				
Tests Blocked	0	0	5	5	3	3	10	10	10				
Execution Plan	20	20	45	45	50	50	70	70	80	90	100	100	100
Total Executed	20	20	45	45	50	50	70	70	80	0			
Test Set Total	20		40		50		50		50				
Test Set Passed	5		15		32		20		20				
Test Set Failed	15		25		15		35		20				
Test Set Blocked	0		5		3		10		10				
Test Set Name	1.00		1.01		1.02		1.03		1.04				

Rallying the Test Organization

The final key focus area within Test-Driven Project Management is rallying the test organization. How do you maintain a high level of energy within the test team? How will your team deal with adversity? And how can you reward team members outside of the annual performance review?

Motivation

After the initial excitement of launching a project and serving as the focal point for project status, there will be down periods where it will feel as if the team's effectiveness as a driver has worn off. Generally as the state of the product improves to beta quality, the focus of the project will shift to that first beta customer. An opportunity that should be explored with your test team is to be more involved with the actual beta trial or maybe volunteering to demo the product in your lab to potential customers. This gives your team an opportunity to learn more about how the customers are actually using the product and should also instill pride when the demo or beta trial is a success. As a reward for being so proactive throughout the project, this offer will most likely be welcomed by both the customer service and sales organizations.

Dealing with Adversity

Anytime change is introduced in a company there will be resistance. Given how our Test-Driven Project Management effort was launched, our first challenge was making sure that the development and test organizations were working effectively as a team. After flooding the development team with defect reports on the initial submissions, the test team found itself spending more time initially working product launch issues such as documentation and packaging rather than executing tests. Unfortunately, while the test team had successfully reduced the test cycle duration by becoming more proactive, the development team felt attacked and viewed the change in an unfavorable light. The Development Management team was then put on the hot seat at the Senior-Manager meetings and was consistently asked how they could improve their process. Luckily, the end result of the meeting helped spark the idea for the prioritized defect report described earlier, which allowed the Development team to greatly increase their defect resolution time. The lesson learned from that situation was that any setback or attack could quickly be turned back into another opportunity to excel.

Rewards & Recognition

While monetary rewards are always readily accepted, they are not always available. And even when bonuses become the norm again, being recognized by the Company for a job well done is what motivates people to come to work and stay loyal. As you work your way into the Senior Management meetings make sure you take the opportunity to sell your team and ensure they are recognized at the next quarterly review. Use their names when describing test efforts at these meetings and introduce them to the same executives you are presenting to. Simply sharing a compliment received from senior-management with your team can make someone's day.

Summary

It has been roughly one year since we first started implementing a Test-Driven Project Management process. By following this system, the test team members have gained: visibility throughout the organization, credibility from departments who had previously viewed the team as pessimistic, popularity, in that, their feedback and participation is actively sought out when planning new projects and confidence that they can influence changes within the company.

While we cannot say for certain that the product delivery process has substantially improved yet, the experience gained and contacts made in the past year have surpassed what most of the team had taken a career to obtain.

References

Demarco, Tom and Lister, Timothy. "Peopleware: Productive Projects and Teams, 2nd. Edition", 1999.

Demarco, Tom: "The Deadline: A Novel about Project Management", 1997.