Aligning Technology Transfer Using Basic Business Measures
David Dorenbos
Motorola
13013 E. Algonquin
Schaumburg, IL 60196
18475760421
Dave.Dorenbos@Motorola.com

ABSTRACT
Motorola Software and System Engineering Research Lab has 15 years of experience in technology transfer. The business priorities and measures of success in technology transfer are discussed and an example is given.

Categories and Subject Descriptors
D.2 Software Engineering, D.2.8 Software Metrics

General Terms
Management, Measurement, Economics

Keywords
Software Engineering, Technology Transfer, Software Metrics, Business measures, P/L Statement

1. INTRODUCTION
Varied understanding of success measures and of the time needed to achieve the desired final improvement often hinders technology transfer from university research to industrial use. Understanding of the success measures and time needed to demonstrate success in the eyes of all the participants could help us to plan technology transfer that provides measurable benefit.

2. PARTICIPANT CONFLICT IN EXPECTATIONS
When a technology moves from a university to a business, there are varying concerns among the participants. The university participants question is often: “Is the technology introduction a significant advance in the state of the art?” Software engineers responsible for the transfer ask, “Is the technology supported such that issues, problems and opportunities can be rapidly resolved?” Software engineers using the technology ask, “Will using this technology provide immediate improvement in my productivity and quality of work?”

The business owner or general manager asks the overriding question “Is this the best investment I can make to improve the return on the products we make?” Donald J. Reifer pointed out that engineers in industry very often do not consider the business case for technology change. “In my experience, few engineers present credible business cases. Instead, they examine the engineering tradeoffs and prepare their arguments around the technical merits of the case. To survive in the business world, engineers must provide others with the business case information they are looking for to make decisions[1].”

3. THE ULTIMATE MEASURES OF SUCCESS, THE P/L
The strongest business case is one that makes substantial impact on the income statement (or profit and loss statement, P/L) of a business as all industrial income and costs are gathered together in the P/L. If you are not familiar with the income statement, many web sites provide an overview. The U.S. Securities and Exchange Commission website has a concise explanation, “Beginners' Guide to Financial Statements.”[2]

Affecting the P/L of any business requires a significant effort but affecting a large business is a special challenge. Consider the fourth quarter P/L for Motorola [3] shown in Figure 1.

An ideal software engineering technology transfer has substantial impact on Net sales, lowers the Cost of Sales through decreased effort accrued to an individual product, lowers Selling, general and administrative (SG&A) expenses by lowering warranty costs and lowers Research and development expenditures by decreasing the effort to create and test future products. For those who have made software engineering technologies a career, cost of quality dollars or productivity per staff may be more familiar but we should not forget that our measures must have impact on the P/L if we are to have significant effect on the businesses that employ us.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

"TT'06, May 22, 2006, Shanghai, China.
Copyright 2006 ACM 1-59593-085-X/06/0005...$5.00.

19
Condensed Consolidated Statements of Operations
(In millions, except per share amounts)
Quarter Ended
December 31, 2005

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net sales</td>
<td>$10,433</td>
</tr>
<tr>
<td>Costs of sales</td>
<td>7,142</td>
</tr>
<tr>
<td>Gross margin</td>
<td>3,291</td>
</tr>
<tr>
<td>Selling, general and administrative expenses</td>
<td>1,145</td>
</tr>
<tr>
<td>Research and development expenditures</td>
<td>916</td>
</tr>
<tr>
<td>Other charges/(income)</td>
<td>(519)</td>
</tr>
<tr>
<td>Operating earnings</td>
<td>1,749</td>
</tr>
<tr>
<td>Interest income (expense), net</td>
<td>55</td>
</tr>
<tr>
<td>and businesses</td>
<td>(69)</td>
</tr>
<tr>
<td>Other</td>
<td>(15)</td>
</tr>
<tr>
<td>Total other income (expense)</td>
<td>(29)</td>
</tr>
<tr>
<td>Earnings from continuing operations before income taxes</td>
<td>1,720</td>
</tr>
<tr>
<td>Income tax expense</td>
<td>510</td>
</tr>
<tr>
<td>Earnings from continuing operations</td>
<td>1,210</td>
</tr>
<tr>
<td>Loss from discontinued operations, net of tax</td>
<td>(8)</td>
</tr>
<tr>
<td>Net earnings</td>
<td>$1,202</td>
</tr>
</tbody>
</table>

Figure 1 - Motorola 4th Quarter 2005 Profit and Loss Statement

4. Solutions competing for impact

When people responsible for a Profit and Loss Statement want to improve the Net earning line they will consider competing strategies not all of which have software engineering as a primary component of improvement as shown in Table 1 - Competing Solutions to Business Problems, Table 1.

Table 1 - Competing Solutions to Business Problems

<table>
<thead>
<tr>
<th>Solution</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
</tr>
</thead>
</table>
| Hiring more staff and hiring cheaper staff may have impact on software engineering strategies but they are not primary opportunity for introduction of new technologies. Rather, the organization will struggle to impart knowledge of current processes and technologies to new staff. Hiring more staff is a means of producing more product, of course with proportional increases in direct engineering and in research and development cost. Hiring cheaper staff will again cause the organization to struggle with imparting domain knowledge. The end-goal to produce to same products but with lower direct cost and R&D engineering cost. However, both strategies, successfully applied in one three-month period can begin to show significant impact on the quarterly P/L in the next 3-month period. On the other hand, software engineering technology transfer will usually take years to affect the P/L.

5. Time to impact

The Ericsson experience in technology transfer was discussed in the July/August 2004 IEEE Software magazine [4]. A key observation was the number of iterations in technology transfer that are necessary to increase the degree of adoption in engineering practice. For high success in transfer of technology, at least three iterations of transfer activities were needed. Analysis and design technology took the greatest investment of time as full five iterations with a 6th started.

For Motorola, technology transfer iterations align to product release. A business may release product every three to six months but the work on a release may take one to two years with multiple releases developed simultaneously. If proof of a technologies value within a release cannot be demonstrated until the product is delivered to the customer, wide adoption of the technology is slowed. Wide adoption will not happen until potential users are convinced that there will be significant impact on their release. If the value of a software engineering technology is not measurable until after a product ships to customer then one iteration cycle for technology transfer activities can be one to two years. In a large organization with multiple products and multiple release of each product it can take 4 to 10 years before P/L impact is measurable.

P/L impact taking years is confirmed by other software engineering research initiatives as shown in Table 2.

Table 2 - Motorola Research Initiation Durations

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Research Start</th>
<th>First Pilot</th>
<th>Division P/L Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian Networks</td>
<td>2001</td>
<td>2002</td>
<td>2005</td>
</tr>
<tr>
<td>Regression Test Selection</td>
<td>2000</td>
<td>2001</td>
<td>2004</td>
</tr>
<tr>
<td>TTCN3 define &amp; adopt</td>
<td>2000</td>
<td>2004</td>
<td>2006</td>
</tr>
<tr>
<td>UML 2.0 co-simulation</td>
<td>2004</td>
<td>2005</td>
<td>2006</td>
</tr>
</tbody>
</table>

6. Impact example

In the early 1990’s Motorola Labs began a research program in code generation. While the code generation technology was innovative and an advance in the state of the art we had learned that creation of models to drive the code generator was a major roadblock. We had previous experience in modeling of requirement to early validation via simulation. That experience had taught us that creating, maintaining and expanding the model
editing tool required more resource than a small lab team to apply. Therefore, when we needed a modeling language that could provide comprehensive input to a code generator we selected SDL96 and later SDL2000.

In the mid-1990s we began using code generation driven by SDL models created in a commercial SDL tool to build shipping software product. This introduction had strong management backing with the general manager convinced that adoption of this technology was an opportunity to lower the R&D cost, lower the direct engineering cost and lower SG&A through improved quality. Through the mid to late 1990s we acquired data that demonstrated 3 to 10 X improvement in quality relative to similar organizations and industry benchmarks. Similarly, we could show dramatic improvement in productivity. However, both the measures were on a “per team” basis and while we could extrapolate to business-wide use of the technology we had no actual data that demonstrated effectiveness.

Finally, in 2000 there was broad spread use of SDL and code generation was segregated by release as shown in Figure 2,

![Product Releases with 1000 R&D staff (60% software)](image)

Figure 2 - Productivity per release with and without SDL usage

Three small releases had no new SDL content; all the work was on a small amount of code. Two of these releases had lower productivity than the SDL releases while one very small release had very high productivity. However, when the staff months for the SDL releases were summed and divided it into the sum of the KAELOC (a KAELOC is one thousand assembler equivalent lines of code) we saw a productivity of 1.37 KAELOC per staff-month and compared .64 KAELOC per staff-month for the releases with no SDL content. The count of staff-effort included the 40% of the staff that had role in creating software but were never the less an integral part of the costs of the division. This productivity difference could then be translated into the additional staff needed to create the large releases if SDL and code generation were not used. The calculation showed that 324 additional staff would be needed and staff increase would mean an increase in R&D cost of 30%. Furthermore, when companies are benchmarked for Sales/R&D-employee[5] a 30% staff increase will move a company from a leadership position to a record of poor performance.

7. Lessons for university to industry technology transfer

Universities attempting to interest industry in a technology change would do well to decide which of the top-level P/L lines the innovation is targeting. Once the relevant P/L targets are selected the university researchers should attempt to build a business case that demonstrates the impact of the innovation; the outcome may be to sharpen the marketing of the technology.

If the technology falls into the class that will require several iterations of adoption to show business-wide impact then the university researchers should think through intermediate measures. One well known means of arriving at intermediate measures is the Goal-Question-Metric paradigm as suggested by Reifer.[6]

Software engineering teams likewise should understand the P/L lines to which their company has the most sensitivity. For instance, if increasing sales relative to competition is difficult then reducing direct and R&D engineering cost may take priority.

A discussion of the P/L target between the university research team and the industrial receiver is vital to moving a technology transfer from the category of the merely interesting to broad impact on a company and an industry.

References