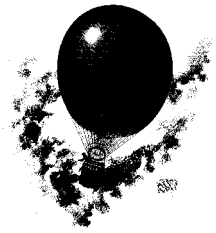
Sunday, Oct. 15, 2006
Vol. 22, No. 19

Soft•letter

BUSINESS INSIGHTS FOR SOFTWARE DEVELOPERS & PUBLISHERS

The Methodologies and Economics of Updating

By Bill Wise, PocketSoft



Operating income per employee figures are rising in our latest survey. Also, we've added a new SaaS category
See pages 4-5.

In the past, patches were perceived as a good thing—a permanent fix or enhancement for applications, that could be delivered with minimal bandwidth requirements, and no reinstallation was required. But today the term "patch" has taken on new, often erroneous connotations. Patch has now come to mean a security response to malicious hacking of operating systems and their associated software. Today patch not only has a negative connotation, it has become an almost meaningless term to many because it is currently used to describe several very different update methods. I'll therefore continue this discussion using the term "update" instead. An update includes bug fixes, product enhancements, minor and major feature additions, documentation improvements, data revisions, etc. Update methods documented in this discussion are all forms of lossless compression (i.e., the files are 100% identical before and after transmission).

Common Update Methods

When faced with the task of distributing an update to customers, software companies usually use one of four common update methodologies: 1) Create new installs for each new version; 2) Release "partial" (aka incremental) installs that install only the files that have changed since the last release; 3) Use "block-level differencing" to extract only the changed blocks for each file that needs to be updated; or 4) Use "byte-level differencing" to process all files to be updated by extracting and encapsulating a sequence of byte-level instructions to transform the old files into the new files. Depending on the software to be updated, each update method may have distinct advantages or disadvantages.

Method 1: New Installs for Each Release

Creating new installs for each release has the advantage of being simple for the update package developer since there are no additional steps required beyond the creation of the primary installation. Each install is stand-alone, so you need not worry about which of your software versions the customer may already have installed. But there are several significant disadvantages to full reinstallation as an update method: 1) The customer bears the burden of uninstalling previous versions and installing the *(continued on page three)*

Publisher & Managing Editor
Merrill R. Chapman
rickchapman@softletter.com
860/663-0552

Editor
Donald K. Rosenberg
don@softletter.com
919/687-4172

Editor Emeritus
Jeffrey Tarter
jtarter@softletter.com
617/668-0028

Editorial office
Soft•letter
34 Sugar HIll Rd.
Killingworth, Conn.
06419
860/663-0552

Subscription office
United Communications Group
11300 Rockville Pike
#1100
Rockville, Md. 20852
301/287-2718
866/313-0973
customer@softletter.com

www.softletter.com

The Latest in Search Engine and AdWords Shenanigans, Part III of III

Social Bookmarking Sites and Tag and Ping Techniques

Social bookmarking, exemplified by such sites as Technorati and Del.icio.us, is the latest online rage, with software companies such as 37signals (SaaS provider of Basecamp) leveraging their power to build sales. Different social sites are aimed at different site types; Technorati, for example, primarily ranks and examines blogs and to maximize your investment in social bookmarking, consider creating a blog.

Social sites allow their users to collect and track bookmarks in public folders which are shared with other site visitors. The bookmark can be tagged with keywords such as "Windows," "politics," "manufacturing," etc. Regular visitors to the social systems can visit popular sites rated by tag activity, subscribe to RSS feeds, and search for tags (keywords) of interest to them. Over time, as your bookmarks becomes more widely tagged under relevant keywords, your site or blog will rise in popularity on the social sites with the result that more visitors will swing by your URL to pay you a visit. An important side benefit of performing well on the social sites is that the major search engines are paying increasing attention to their rankings.

Companies leverage the social sites by first analyzing what keywords are most relevant to their market, then submitting their own URLs to the social sites for review. The most popular technique for ensuring your site is indexed by the social sites is via "tag and ping" campaigns. These consist of:

- Creating content on your site or blog, then adding tags at end of the post.
- Mounting your content on a site.
- Pinging the social bookmarking sites. Pinging is a automated notification to a social website (though the technique can be used with any site) informing it that your site has been updated.)

The following is an example of a tag designed for Technorati:

" [tagname]

The [tagname] includes keywords under which you want to be ranked. For example, if your company is selling budgeting software, you might want to have the site rank you on the keywords "budget" and "rollup." Your Technorati tag would look like this:

" budget rollup

(The "+" lets you include more than one keyword in a tag.) Blog tools such as WordPress can automate the process of tagging and pinging considerably.

Do not, under any circumstances, attempt to artificially inflate the number of pings and tags for your site via spam techniques; when caught, your site will be thrown out of the bookmarking site's rankings permanently.

new version; 2) If the new installs are delivered via ftp or web sites, bandwidth is wasted since the full install must be downloaded for each new release; 3) If you're sending the full install electronically, you'll probably want to compress it and ordinary compression using formats such as ZIP or CAB can at most only achieve 40 to 60% size reductions; 4) If media (CD/DVD) is used to distribute the new installs, shipping and handling costs can be significant. For situations where new versions are released only once every few years, full installs make sense. For more frequent release schedules, an easier and more efficient update method is necessary.

Method 2: "Incremental" Installations

Instead of releasing full installs, some companies use "incremental" installs, again using ordinary compression. By identifying only the files that have changed, the software update is reduced in size, the amount depending on the number and size of files that have changed. For example, if only a few 10KB sized files have changed since the last release, the incremental update will be small. But if the changed files are 50MB each, the incremental update will grow dramatically. Incremental installs are an obvious and heavily used method of reducing update bandwidth requirements, but they still fall short when bandwidth is narrow. To address this issue, you need to implement **file-level differencing**. In other words, for each file that has changed, determine what information is new, and what information is old. Then create an update that can transform the old version to the new version using only those "new" bytes.

"Companies such as Microsoft use a mix of update systems, including file and byte-level products. For games, byte-level differencing systems such as RTPatch are the standard."

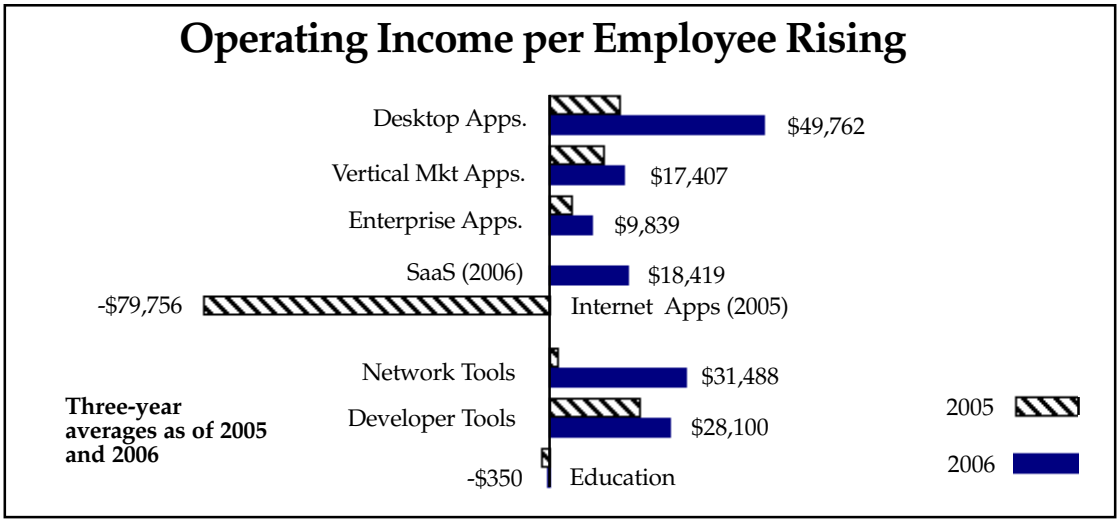
—Bill Wise PocketSoft

For file level differencing, there are two options: block-level and byte-level.

Method 3: Block-Level Differencing

In block-level differencing, changed files are dissected into distinct "blocks" of data. Common block sizes are 16KB and larger. Blocks are then compared to the previous version's blocks. Only the changed blocks are included in the software update. Though it is fast and efficient for some data updates—especially appended data—block-level differencing has two notable disadvantages: 1) It performs poorly on data files with scattered changes; 2) It is virtually useless on compiled files such as executables, DLLs, etc. In both cases, the problem is that even if only one byte has changed, then the entire block must be sent to update the file. In the case of a 16KB block where only one byte has changed, 15,999 bytes (which are already present in the old version) must be resent to the customer. Because of these architectural factors, block-level differencing is ill-suited for software updating. However, software "updating" is not the same as "synchronization".

In its favor is the fact that block-level is better suited for synchronization than updating. Block-level differencing works with varying degrees of efficiency for remote synchronization *(contined on page six)*



Benchmarks: Operating Income per Employee

The table opposite shows operating income (earnings before interest and taxes, or EBIT, also called operating profit) and divides this by the number of employees. The information comes from public filings and other sources. Some firms are enormous and enormously efficient (Microsoft); economies of scale do exist, as the table below indicates:

Median Operating Income per Employee
by Size of Firm's Average Operating Income

Firm Size	Operating Income Ratio
>$100k	$58,274
$51k-99k	$26,230
$10k-$50k	$28,598
$1k-$9k	$16,219

But scale is not everything, and we have made a large change in the Benchmark 50, dropping the Internet sector to include the growing Software as a Service (SaaS) sector. And rather than put Google with its sisters in that sector, where it would distort the sector figures, we have put it up there with Microsoft to indicate how much of its own universe it is. Google demonstrates the power of SaaS: with less than a tenth of the employees of Microsoft, it is generating 35% more income per employee. Google and Microsoft demonstrate that market leadership is very good for profits, and a look at the other sectors demonstrates that having specialized products, that is, dominating a niche, has the same effect on a smaller scale.

The graph above compares the running three-year average Operating Incomes per Employee from last year and this year, from which we can see the returning strength of the software industry. The new SaaS sector weighs in at the median of this ratio, and we can see that the Internet Applications sector was the worst-off of any in last year's study.

Finally, a change in Desktop Applications for the Benchmark 50 reflects the changing market: IMSI merged with AccessMedia Networks, Inc. and took the name Broadcaster, Inc. Thus IMSI joined the SaaS market by providing users with an online set-top box. In IMSI's place we added CyberLink Corp., which uses PCs and consumer electronics to display and to network home entertainment.

The Benchmark 50: Operating Income Per Employee

	Employees			Operating Income/Employees			Avg.
	2003	2004	2005	2003	2004	2005	'03-'05
Microsoft	**57,000**	**61,000**	**71,000**	**$158,491**	**$238,705**	**$232,000**	**$209,732**
Google	**n/a**	**3,021**	**5,680**	**n/a**	**$211,914**	**$355,155**	**$283,534**
Desktop Applications				**$32,086**	**$71,475**	**$45,725**	**$49,762**
Intuit	6,700	7,000	7,500	$62,609	$74,871	$74,606	$70,695
Macromedia	1,085	1,213	1,445	$1,562	$39,610	$38,649	$26,607
Adobe	3,507	3,142	5,734	$75,946	$143,348	$106,355	$108,550
Symantec	5,300	6,500	16,000	$96,903	$126,041	$17,123	$80,022
Cyberlink	n/a	290	310	n/a	$68,079	$75,955	$72,017
Nuance Commo. (Scansoft)	472	n/a	1,112	-$13,691	n/a	$1,827	-$5,932
Smith Micro	55	52	91	-$17,400	$66,596	$45,725	$31,640
Vertical Market Applications				**$13,797**	**$21,035**	**$17,389**	**$17,407**
Autodesk	3,493	3,477	4,813	$30,404	$67,558	$76,834	$58,265
Moldflow	291	318	331	$10,148	$21,035	$553	$10,578
Ansys	610	550	600	$49,700	$83,596	$98,067	$77,121
Advent	764	752	736	-$51,910	-$24,021	$8,609	-$22,441
Dendrite	2,524	2,549	2,793	$13,797	$18,762	$12,955	$15,171
MapInfo	708	800	903	-$2,942	$13,374	$17,389	$9,273
Kronos	2,400	2,500	2,900	$22,734	$27,841	$28,117	$26,230
Enterprise Applications				**$2,454**	**$14,148**	**$12,916**	**$9,839**
Concur Technologies	313	350	395	$2,454	$4,997	$12,916	$6,789
Manhattan Associates	1,117	1,400	1,600	$27,300	$22,578	$18,923	$22,934
Knova Software (ServiceWare)	51	106	95	-$19,728	-$157,225	-$56,286	-$77,746
Pegasystems	422	406	458	$34,773	$14,148	$1,116	$16,679
Witness Systems	497	485	619	-$43,390	$17,633	-$2,485	-$9,414
SPSS	1,252	1,165	1,129	$102	$5,645	$24,822	$10,190
Business Objects	3,924	3,834	4,418	$9,989	$21,323	$29,922	$20,411
SaaS				**$23,765**	**$10,332**	**$21,161**	**$18,419**
Salesforce.com	n/a	767	1,304	n/a	$8,501	$15,416	$11,958
RightNowTechnologies	n/a	403	530	n/a	$8,444	$12,311	$10,378
Digital Insight	741	761	800	$23,765	$37,039	$50,334	$37,046
WebSideStory	n/a	135	214	n/a	$12,163	$26,907	$19,535
Savvis	993	1,858	2,124	-$87,187	-$51,910	-$1,731	-$46,943
WebEx Commo.	1,241	1,826	2,091	$34,731	$38,040	$38,223	$36,998
Network Tools				**$17,406**	**$32,972**	**$44,086**	**$31,488**
Citrix Systems	1,885	2,656	3,171	$81,904	$59,861	$64,488	$68,751
McAfee (Network Associates)	3,700	2,950	3,290	$17,406	$109,380	$48,064	$58,283
iPass	330	402	406	$67,748	$72,055	$44,086	$61,296
Novell	5,734	6,000	5,100	-$4,852	$10,949	$84,831	$30,309
Altiris	600	750	878	$21,305	$32,972	-$3,485	$16,931
Tumbleweed	133	270	320	-$74,511	-$29,496	-$15,213	-$39,740
NetManage	236	225	204	-$19,758	$6,698	$11,926	-$378
Developer Tools				**$26,421**	**$25,054**	**$32,824**	**$28,100**
Raining Data	142	143	133	-$1,324	$1,441	-$3,917	-$1,267
Pervasive Software	245	251	165	$31,718	$14,805	$7,152	$17,892
Progress Software	1,391	1,552	1,593	$26,421	$29,874	$40,028	$32,108
Borland Software	1,358	1,361	1,269	-$29,406	$13,721	-$27,561	-$14,416
Sybase	3,660	3,568	3,715	$28,505	$25,054	$32,824	$28,794
Red Hat	681	940	1,100	$4,326	$28,666	$52,802	$28,598
BEA Systems	3,122	3,353	3,878	$55,957	$58,118	$53,078	$55,717
Education				**$9,775**	**-$112**	**-$10,712**	**-$350**
Renaissance Learning	988	942	956	$52,757	$41,394	$35,444	$43,198
Apollo Group	8,036	10,487	11,302	$48,191	$41,787	$63,073	$51,017
Saba Software	258	334	516	-$47,946	-$9,219	-$12,362	-$23,176
American Education Corp.	73	70	57	-$10,902	$18,224	-$10,712	-$1,130
Scientific Learning	130	160	173	$22,569	-$4,594	$30,682	$16,219
Plato Learning	750	787	598	-$3,020	-$112	-$45,841	-$16,324
SumTotalSystems	n/a	447	628	n/a	-$35,725	-$17,033	-$26,379
All companies (median)				**$4,244**	**$30,911**	**$21,230**	**$18,795**

Note: ""Years" may not correspond to company fiscal years.

of data files. As an example, assume that you have a 100MB file (i.e., the "old" file) located on a client machine in Tokyo that you want to synchronize with a file that resides on a server in Houston (i.e., the "new" file). Block-level differencing enables efficient remote synchronization by instructing the client in Tokyo to send a file containing only the old data file's "block signatures" to the server in Houston. The server in Houston can use those block signatures to identify which blocks in the old file differ in the new file. By sending only the changed blocks back to the client in Tokyo, bandwidth is conserved during the synchronization. rsync, an Open Source utility, is one example of remote synchronization based on block-level differencing.

Method 4: Byte-Level Differencing

For software updates or other situations where block-level differencing is not well suited, "byte-level differencing" is a further improvement in efficiency. Like block-level, byte-level differencing examines two versions of a file and extracts only the changed portions, but with byte granularity, rather than block granularity. Byte-level encodes instructions to update files with reordered data, scattered changes, and complicated change patterns. This gives byte-level a clear advantage, especially in the case of updating executable files and data with scattered changes, neither of which are amenable to the block-level approach. The difficulty lies in identifying the changes and encapsulating them in an efficient, effective and reliable manner. A word of caution is in order here. In today's increasingly complex computer environment, reliability of byte-level differencing algorithms can only be established by constant, massive and worldwide usage. RTPatch, a commercial application, is one example of a software update solution based on byte-level differencing.

Case Study: National Geospatial-Intelligence Agency

Global Positioning System (GPS) can tell you where you are in the world, but unless coupled with map data, GPS can't tell you anything about a location’s landscape. In the case of ships at sea, the near-land topography of the ocean floor is constantly changing due to storms, earthquakes and the introduction of man-made obstacles. In the past, the ocean floors had been charted with over 5,000 paper charts which were manually updated, causing them to be frequently out-of-date, often with disastrous consequences. The National Geospatial-Intelligence Agency (NGA) launched a program to digitize those charts. Once digitized, manually updating paper charts could be eliminated in favor of electronically distributed updates. The NGA needed to efficiently distribute those updates and implemented an in-house system built around block-level differencing and partial installations. But the update system could not handle the types of changes inherent to map data (scattered changes, massive amounts of data).

After years of failed attempts to scale up the existing software, the entire program was in jeopardy. Tacking to another course, NGA decided to implement an out-of-the box byte-level solution. Designed to handle precisely this type of update situation, the new system was able to solve the update problem quickly, efficiently and at a fraction of the cost of the prior system. Today that solution has been in place for half a decade, over 6,000 distinct sea bottom changes are updated monthly, and the US and Royal Navies, as well as commercial fleets, rely on those software updates to safely navigate sea lanes. After dozens of man-years and countless dollars, something that many software companies consider as an afterthought—the update process—nearly scuttled the entire project. By matching the right update technology to the right situation, the NGA was able to save the project and score a major technology success.

Bill Wise, CEO, PocketSoft, 7676 Hillmont St., Ste. 195, Houston, Texas 77040; 713/460-5600. E-mail: wise@pocketsoft.com. Website: www.pocketsoft.com.

Representations and Warranties Insurance

By Ward Carter, Corum Group

When working with sellers of privately held software companies on an M&A transaction, one issue that always causes concern is the concept of seller's indemnification and the liability that goes with that. Liability for most items is generally limited to a predefined escrow reserve of 10% or 15% of the purchase price. When issues arise after the close, that reserve can be used to satisfy claims without the buyer having to chase selling shareholders. But, some claims, such as challenges regarding intellectual property ownership, are often only limited to the total of the transaction amount. After 15 years spent building a business you've just sold, the idea of having to return that money is an unpleasant thought. A solution may be to shift the risk to a third party through reps and warranties insurance.

These policies are only available through a few of the bigger insurance carriers, and are not the types of coverage your local agent will routinely offer. And they can be expensive, ranging from 3% to 5% of the coverage, depending on the risk perceived by the insurance company. On a hypothetical deal, say for a $20,000,000 transaction, the escrow might be $2,000,000, and offer the first line of defense against claims by the buyer. If a total cap of $12,000,000 was negotiated, insurance for the remaining $10,000,000 of potential liability might cost $300,000 to $500,000, buying peace of mind (at a cost) for the seller.

There is also a benefit to the buyer if the buyer is the named insured; in the event of claims above the escrow, the buyer can make a direct claim against the policy, which is preferable to a policy where the seller is the insured and the seller must make claims on the policy.

Beyond cost, the addition of insurance does add another layer of complexity and negotiations, as the insurer needs to perform due diligence in an effort to assess and limit their exposure. This requires a measure of cooperation between buyer and seller. I also know of buyers obtaining insurance to cover potential losses without the seller's knowledge. This is a tool used by some savvy private equity firms to manage their risks.

Ward Carter, president, Corum Group, 10500 NE Eighth St., Bellevue, Wash. 98004; 425/455-8281. E-mail: wcarter@corumgroup.com.

Company/Description	Acquired by	Price/Terms	Revenues	Multiple
Catalyst Enterprises *• Analysis and emulation tools*	LeCroy Corporation	$33,500,000 *Terms: Cash*	$10,400,000	**3.22**
Citadel Security Software *• Security risk management*	McAfee (MFE)	$56,150,000 *Terms: Cash*	$14,240,000	**3.94**
YouTube *• Online video-posting*	Google (GOOG)	$1,600,000,000 *Terms: Stock*	<$1,000,000	**OOS**[1]

1. *Out of sight. The YouTube purchase represents a classic strategic purchase.*

CardFile

Book Review: *Innovation Games* by Luke Hohmann, Addison Wesley, 2006

In 2004 Softletter ran a two-part series of articles on Luke Hohmann's innovative series of game techniques designed to help high-tech and software companies better understand your sales team's needs, gain insight into customer needs, and understand complex product relationships. The ensuing book that's resulted from Hohmann's earlier work is fun to read, provides fascinating techniques for working with both customers and your company, and comes at this much discussed topic from a decidedly different viewpoint.

Some of the techniques described in ***Innovation Games*** include Product Box (ask your customers to create a box describing your software's most exciting features), Speed Boat (visualize the anchors holding your product back), Give Them a Hot Tub (encourage your customers to use cognitive dissonance to develop product breakthroughs) and similar techniques. The book includes templates, "trading cards" keyed to each game/exercise, and guidelines on how to run and facilitate sessions. As readers of **Softletter** know, we think the use of game techniques and technology in marketing and sales is underutilized. ***Innovation*** receives our highest rating.

BLOGGER BILLY MARSHALL ON VMWARE: "Last week, Intel announced that it plans to certify its motherboards to run VMware's operating system. The VMware operating system is marketed under the brand name VMware Infrastructure, but it is effectively a "hardware" operating system....

'Why is this important?' you might ask. Well, it is important because historically applications were artificially bound to the hardware infrastructure because the standard operating system had to support BOTH the hardware AND the application. Separating the operating system into independent 'hardware' operating systems and 'application' operating systems allows the hardware infrastructure to evolve independently from the application infrastructure, and vice versa. With this announcement, any application running any version of any operating system can be wrapped in a virtual machine container (creating a virtual appliance) and deployed to an Intel server by simply copying a file onto the "hardware" operating system." (Quoted on http://billyonopensource.blogspot.com/2006/10/standard-os-is-virtually-gone.html, 10/07/2006)

ZDNET BLOGGER DANA BLANKENTHORN ON OPEN SOURCE PROFITS: "Mass market applications require scale, organization, and hierarchies, in order to deliver ongoing support, especially against security threats. The structure of most open source enterprises have yet to scale. The question is, will they ever?

I don't have an answer to that, except to outline a possible business plan. Corporate Office. We take OpenOffice, we hire top committers to it, and we sell support packs to businesses at 10% of what they're paying for the other guys. We donate the templates and other code we create to the community. We make a ton of money." (Quoted on http://blogs.zdnet.com/open-source/?p=803&tag=nl.e539, 10/04/2006)

Soft•letter is published 24 times per year; ISSN: 0882-3499.

Subscription rates: $395 worldwide. Subscription office: United Communications Group, 11300 Rockville Pike, #1100, Rockville, Md. 20852-3030; tel 301/287-2718 866/313-0973 customer@softletter.com

www.softletter.com