Chapter XV

Software Systems for Project Management

You miss 100% of the shots you don't take.

(Wayne Gretzky)

A vast amount of project management software is available today in a wide variety of capabilities, applicability, platform requirements, and prices. These software products significantly enhance the PM's job of managing a project in almost all aspects, including selection, planning, scheduling, execution, control, risk, and communications. PMs should therefore be aware of the types of tools available and the features and applicability of those tools. In this chapter, types of software products and some specific products are identified and discussed.

Spreadsheets

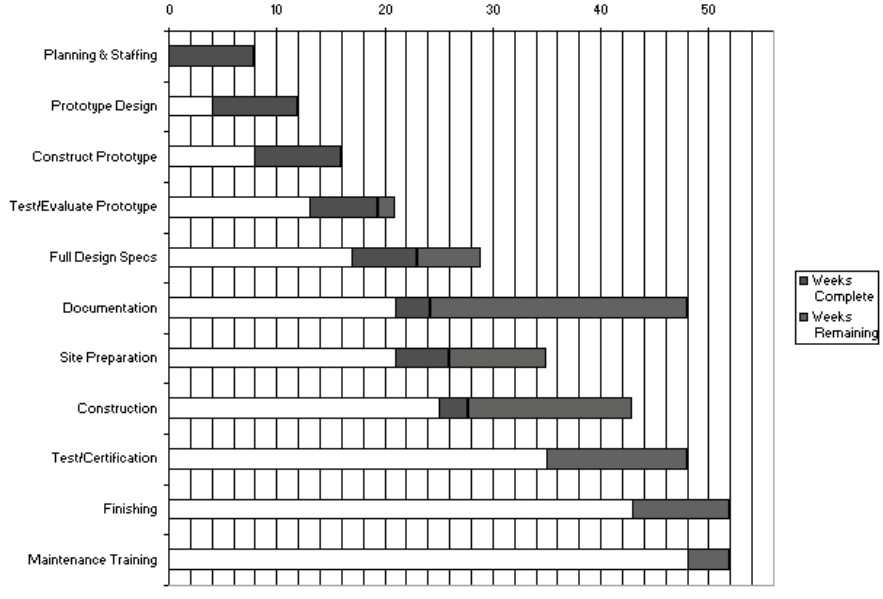
Spreadsheet programs are the most commonly used computer software programs for project management (and business in general). Spreadsheets are easy to learn, easy to use, inexpensive, generally available, and adaptable to most project management tasks. Consider the project planning and progress information shown in the spreadsheet of Figure 15.1; the last two columns are calculated columns. A graphical representation in the form of a classical Gantt chart can also be created from the spreadsheet data in the first three columns using the charting capability in most modern spreadsheet programs (such as Microsoft Excel). In addition, one can also use the columns for weeks completed and weeks remaining to draw a Gantt chart showing progress, as illustrated in Figure 15.2.

Figure 15.1. Project plan and progress

Project Plan and Progress - Week 30

Task	Start Week	Weeks	% Complete	Weeks Complete	Weeks Remaining
Planning & Staffing	0	8	100	8	0
Prototype Design	4	8	100	8	0
Construct Prototype	8	8	100	8	0
Test/Evaluate Prototype	13	8	80	6.4	1.6
Full Design Specs	17	12	50	6	6
Documentation	21	27	12	3.24	23.76
Site Preparation	21	14	36	5.04	8.96
Construction	25	18	15	2.7	15.3
Test/Certification	35	13	0	0	13
Finishing	43	9	0	0	9
Maintenance Training	48	4	0	0	4

Figure 15.2. Gantt chart



The Excel chart wizard was used to create Figure 15.2 with the columns for task, start week, weeks complete, and weeks remaining; the first part of each bar was made the same color as the chart background (the chart is shown in black and white here, but the completed and remaining parts of each bar are different in color). There are also spreadsheet templates available from a number of sources to facilitate Gantt-chart creation.

A cost-based Gantt chart is shown in Figure 15.3. The bars are represented as a series of cost numbers for each time period of each task. This provides for a nonlinear distribution of cost (and resources) across the time span for the task and allows one to quickly see at a glance both the time for a task and the cost distribution (cash flow).

Figure 15.3. Project cost plan spreadsheet

Project Cost Plan

	Ja	Fe	Ma	Ap	Ma	Jn	Jl	Au	Se	Oc	No	De	Total
Planning & Staffing	3	2											5
Prototype Design		3	3										6
Construct Prototype			8	8									16
Evaluate Prototype				5	10								15
Full Design Specs					5	6	3						14
Documentation						2	2	1	1	1	1		8
Site Preparation						8	3	3					14
Construction							20	50	50	20			140
Test/Certification									10	6	4		20
Finishing											8	4	12
Maintenance Training												4	4
Monthly Cost	3	5	11	13	15	16	28	54	61	27	13	8	254
Cumulative	3	8	19	32	47	63	91	145	206	233	246	254	

Figure 15.4. Project cost plan graph

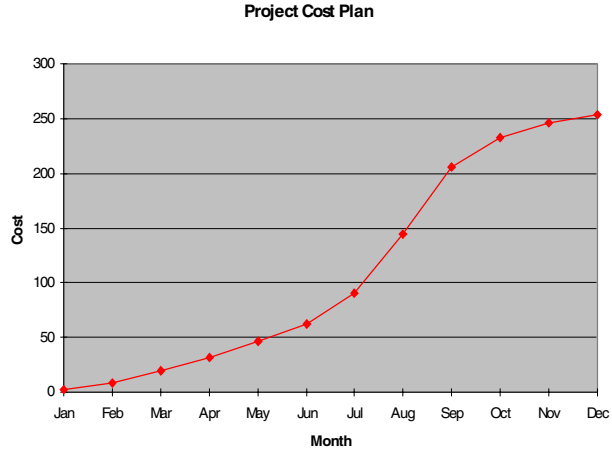


Figure 15.5. Project network diagram

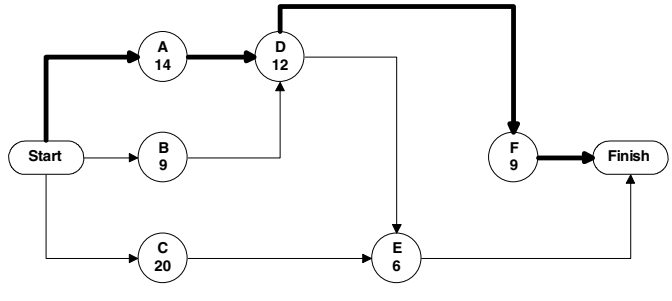


Figure 15.6. Project task information

Task	Duration	Predecessors	Early Start Time	Late Finish Time	Total Slack	Free Slack	Critical Path ?
START	0	---	0	0	0	0	Yes
A	14	START	0	14	0	0	Yes
B	9	START	0	14	5	5	No
C	20	START	0	29	9	6	No
D	12	A,B	14	26	0	0	Yes
E	6	C,D	26	35	3	3	No
F	9	D	26	35	0	0	Yes
END	0	E,F	35	35	0	0	Yes

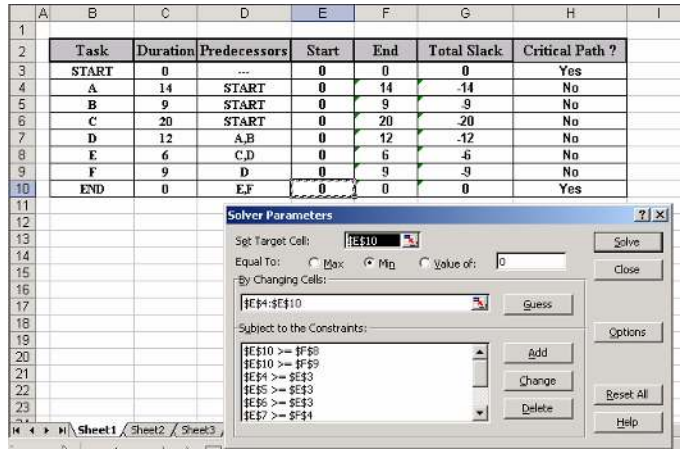
Using charting capabilities built directly into the spreadsheet products, one can use the project cost plan in Figure 15.3 to draw a cost plan graph as shown in Figure 15.4.

In an earlier chapter, the critical path model (CPM) was discussed including the determination of critical path, early/late start times, early/late finish times, and slack/float. Spreadsheet models can also be used to calculate these variables. Consider the CPM drawing in Figure 15.5: The heavy lines denote the critical path and task names and durations are shown inside the nodes.

Once the critical path is determined, a spreadsheet can be used to determine early and late start dates, early and late finish dates, and slacks. This is shown in Figure 15.6 using the network in Figure 15.5, with an example from Klastorin (2004).

Instead of first determining the critical path from inspection (or an algorithm like dynamic programming, used in most scheduling tools), sophisticated spreadsheet programs can

Figure 15.7. Spreadsheet determination of critical path



also be used to find the critical path via linear programming. This is illustrated in Figure 15.7 for the network in Figure 15.5. Using Microsoft Excel, in the solver dialog box, we have requested minimization of the start of the end task subject to constraints for the dependency relationships (i.e., the start time of a task must be greater than or equal to the finish time of its predecessors). Figure 15.7 shows the initial values in the spreadsheet; columns F, G, H are formulas, and column E has the starting values of the cells to be calculated by the LP solver. The formula for column F (task end time) is the start time plus the duration. The formula for column G (total slack or float) is:

(minimum of successor start times minus task end time) plus (task start time minus max of predecessor end times)

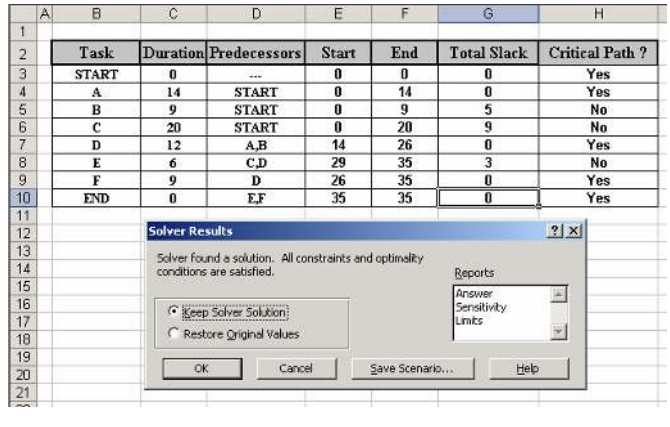
The formula column F (critical path) is "yes" if total slack is zero, otherwise "no."

Figure 15.8 shows the solution. Since the LP criteria was to minimize the end time, the solution started all critical path tasks at their earliest time; however, it also started all noncritical path tasks at the latest times. This can be seen because task E (not on critical path) is started at 29 instead of 26. To force the solution to start noncritical path task at their earliest start time we would minimize the sum of all task start times instead of just the "end" task.

LP solutions are no better in this simple case that regular algorithms in scheduling programs, but if there are other constraints such as cost, cash flow, late finish penalties, or early finish incentives, then a LP solution can handle these additional complexities whereas the standard scheduling programs cannot (Klastorin, 2004).

Throughout this book, spreadsheets have been already been used to illustrate the analysis and solution of problems in a number of project management knowledge areas; Figure 15.9 itemizes these spreadsheet models. There are many spreadsheet add-ons available to facilitate analysis in many knowledge areas. Although spreadsheets are very useful, they are less appropriate for large projects and should not be used for everything.

Figure 15.8. Spreadsheet determination of critical path (answers)



Task	Duration	Predecessors	Start	End	Total Slack	Critical Path ?
START	0	---	0	0	0	Yes
A	14	START	0	14	0	Yes
B	9	START	0	9	5	No
C	20	START	0	20	9	No
D	12	A,B	14	26	0	Yes
E	6	C,D	29	35	3	No
F	9	D	26	35	0	Yes
END	0	E,F	35	35	0	Yes

Figure 15.9. Spreadsheet models in this book

Spreadsheet Model	Chapter
Net Present Value	III
Internal Rate of Return	III
Decision Tree	III
Maximax/Minimax Selection	III
Requirements Specification	III
Organizational Assignment Matrix	VII
Generic Resource Table	VII
Function Point Analysis	VII
Cost Plan	VII, XIV, XV
Risk EMV	VIII
Risk Plan	VIII
Outsource Processes	XII
Outsource Country Analysis	XII
Stakeholder Analysis	XIII
Resource Assignment Matrix	XIII
Earned Value Analysis	XIV
Timesheet	XIV
Schedule & Gantt Chart	XV
Critical Path Analysis	XV
Project Portfolio Optimization	XVI
Project Scoring	XVI

General Project Management Software

There are many general-purpose project management software systems that handle project scheduling and progress reporting. Some of these systems may also include capabilities for some other aspects of project management, including risk. Many project management software systems also handle very specific aspects of project management, such as estimation, risk, change management, portfolio management, earned value, and so forth. Many such specific software programs have already been identified in chapters dealing with those specific aspects of project management.

In addition to features and capabilities, the general-purpose programs may be categorized based on several factors:

- Single or multiuser
- Single or multiproject
- Platform (PC/Mac, file services [redirection], mainframe, client-server, Web based)
- Server operating system (Windows, Unix, Linux, other)
- Data access (file system, ISAM, relational database, other)
- User interface (text, simple GUI, rich GUI, other)

- Import/Export (spreadsheet, XML, etc.)
- Sales/Support (proprietary, open source, shareware, freeware)

The most appropriate package would be a function of capabilities desired, size and complexity of projects, size and complexity of performing organization (number of projects), performance methods used (i.e., earned value or not), platform constraints, location and type of users, and budget constraints. Commonly used general packages include (in alphabetical order):

- @task (www.attask.com)
- Artemis (http://www.artemispm.com/)
- Autotask (www.autotask.com)
- BPSProject (www.bpsproject.com)
- Cooper (www.cooperproject.com)
- EasyProject (www.easyprojects.net)
- EPK-Suite (www.epkgroup.com)
- Microsoft Project & Project Server/Central (www.microsoft.com/project)
- Milestones Simplicity & Professional (www.kidsa.com)
- OpenPlan (www.welcom.com)
- Planview (www.planview.com)
- Primavera & TeamPlay (www.primavera.com/)
- Project KickStart (www.projectkickstart.com/)
- Projistics (www.projistics.com)
- PS Next (www.sciforma.com)
- TeamCenter (www.inovie.com)
- TurboProject (www.tekdeal.com)
- WebProject (www.wproj.com)
- Websystems' AceProject (www.aceproject.com)

The most appropriate software packages for modern IT projects would be Web based, would utilize a comprehensive relational database, are highly scalable, and have a modern GUI. Many of these packages do not have all these features. Some general purpose project management software systems are offered in the form of shareware; that is, one can try the system without charge and then pay a relatively modest fee to continue using the system; examples of shareware systems are MinuteMan, SmartWorks, PlanBee Pro, and QuickGantt.

Open Source Software

Organizations pay significant amounts for annual software license fees; the average Web retailer pays about $300,000 per year. It used to be that free open source software meant "buggy and without support"; the source code itself was the first line of support. However, many organizations are now taking a second look at open source.

There are open source *systems products* such as the Linux operating systems and the Apache Web server, and open source *application components.* The application components include database management systems, application servers, architectures, programming languages, integrated development environments, development tools, browsers, utilities, and various "middleware" components. Both of these types of components may be part of the environment for a project management overall software system. System software products like Linux and Apache are well established now, and their use for prime-time enterprise systems is well documented. For example, United Parcel Service (UPS), which rejected Linux 3 years ago, recently reevaluated this product and is now looking to move all of its RISC-based applications to Linux. Seventy percent of IT executives polled by *Information Week* say their companies now use Linux, up from 56% in 2003 (Greenemeir, 2004). Open source application components are now trying to make the same inroads as has the Open source system software.

The open source model of software distribution and usage is vastly different from commercial products. Open Source software is copyrighted but freely released; extensions and improvements are freely released also. There are a number of free license types, the most popular is the general public license (GNU). Open Source software is free to run, copy, distribute, modify, extend, and improve. Open source software may not have a single product provider. Open source may be developed by an individual, group, or organization and then extended by a "community" of cooperative groups of individuals, each of whom contributes parts of the product. Documentation, support, training, and consulting is generally not centrally available but provided by the community. Some products have sponsor organizations that become focal points for the coordination of official releases of the products and the documentation thereof; most documentation today is online. Sponsor organizations raise funds for their activities via sale of priority support, installation and integration services, other consulting, and/or related products. Sponsor organizations who are also the copyright holder may also release versions of the product under multiple open source–type contracts or under proprietary contracts.

Implementing an Open Source solution requires a different approach from the buyer's perspective. There is no salesperson, no license agreements to sign, and no serial numbers to record. Initial and ongoing license cost savings in using open source software can be huge. For example, the annual cost (per processor) for major vendor relational database management software (RDBMS) follows (Hall, 2003):

- Oracle Standard Edition: $15,000
- Oracle Enterprise Edition: $40,000
- IBM DB2 Workgroup: $7,500

- IBM DB2 Enterprise Edition: $25,000
- Microsoft SQL Server Standard Edition: $4,999
- Microsoft SQL Server Enterprise Edition: $19,999

"After the operating system, it (database management system) is the next significant layer in building an environment based entirely on open source" (Banerjee, 2001). An efficient and scalable relational database product is a key part of a modern general-purpose project management tool that can handle multiple projects. Most major open source products are comparable to their proprietary counterparts in basic features (as prescribed by relevant international standards bodies), and are also comparable in necessary business environment integrity in matters relating to security, backup, recovery, reliability, scalability (multiprocessor, etc.), clustering, replication, and so forth. Thus, a general feature-by-feature comparison of open source versus proprietary products would be meaningless, just as a feature-by-feature comparison of proprietary products is meaningless unless the comparison is with regard to the construction of a specific application for a specific platform.

The MySQL RDBMS is nearly free, depending upon any paid support purchased. In the past, the first line of support for open source products was the source code, but not anymore. For an organization to successfully and economically utilize open source software, it must assemble a coalition of providers for each open source product and use them to create the necessary support infrastructure. Some providers may be paid, such as for priority support (via e-mail, phone call centers, etc.), and others may be Web links to chat sites which link posted problems to proposed solutions. At first glance, it may seem risky not having a single provider responsible for the whole application, but in reality the overall risk is lessened. Major open source products have sponsors with various types of paid support plans, large communities of users who answer problem postings relatively quickly, and a large number of available knowledgeable individual contractors and potential employees. *CIO Magazine* lists (and then dispels) the main myths about open source software (Wheatly, 2004):

- The attraction is the price tag.
- The savings are not real.
- There's no support.
- It's a legal minefield.
- Open Source is not for mission critical applications.
- Open Source is not ready for the desktop.

The development of enterprise-class modern business applications (such as comprehensive project management systems) typically utilizes an application framework that is based upon a particular architecture. Application frameworks are a holistic set of specifications for the interaction and assembly of multiple reusable patterns. A pattern is the design of a core functional element, such as the MVC (model-view-controller) pattern found in many architectures such as J2EE (Java Two Enterprise Edition). Modern

architectures may support one or more application servers, programming languages and integrated development environments (IDEs). The major architectures today are J2EE, Microsoft's .Net, and the Open Source LAMP (Linux, Apache, MySQL, PHP). The boundary between architecture, framework, and programming language is blurry and not the same in different architectures. Examples of modern proprietary application frameworks include IBM's Websphere, Macromedia's ColdFusion and Flex, Sun's I-Planet, and BEA's Weblogic.

Hard benchmarks between open source products and proprietary products are seldom published, and these benchmarks are very problem specific. However, *PC Magazine* performed a reasonably comprehensive benchmark using Java server pages (JSPs) and several RDBMSs; it also wrote the same benchmark using active server pages (ASP) in a .NET to include Microsoft SQLServer in the test (Dyck, 2003). The peak throughout results (in pages per second) for the products were:

- MySQL 4.0 (open source) – 608
- Oracle9i – 629
- Sybase ASE 12.5 – 476
- IBM DB2 Universal 7.2 – 494
- Microsoft SQLServer 2000 SP2– 209

However, the acid test for open systems, both at the system and application levels, has to be Sabre's 4-year project to convert its massive airline reservations and related systems from proprietary mainframe-based system and application software to open systems software (Anthes, 2004). The Sabre systems process more transactions than the New York Stock Exchange—about 50 million transactions per day. Sabre is halfway finished its conversion and reengineering efforts for major applications, and has been successful in utilizing open source products as Linux, Common Object Request Broker, Lightweight Directory Access Protocol, MySQL, and Java. The open source MySQL database runs on a very large server farm of four-way HPrx5670 servers running Linux with about 100 GBs of information. Sabre CTO Craig Murphy stated that "we evaluated several database managers, and MySQL was really the winner from a performance standpoint, and it was certainly the lowest cost." Total cost of ownership is expected to be 40% less with anticipated savings of "tens of millions of dollars" (Babcock, 2004).

Some products are built using open source project management software (i.e., MYSQL), some can operated in an open source environment (i.e., Linux/Apache), and some are distributed completely in an open source manner. The FiveAndDime system utilizes open source software. Some fully open source project management systems are:

- dotProject (www.dotproject.net)
- Double Choco Latte (http://dcl.sourceforge.net)
- GForge (http://gforge.org)
- PHProjekt (www.phprojekt.com)

- ProjectOpen (www.project-open.com)
- Projectory (http://projectory/sourceforge.net)
- ToutDoux (www.gnu.org/software/toutdoux/en/index.html)
- WebCollab (http://webcollab.sourceforge.net)

The FiveAndDime System

Project management software systems are very complex applications. There are many entities involved (work breakdown structures, organization structures, personnel and other resources, time periods, both planned and actual costs, work schedules, etc.) and the database relationships between these entities is complex, including recursive relationships and many-to-many relationships. I have been involved in the design and development of several of these type systems using major RDBMSs such as Oracle, and these have been quite challenging applications to construct. The FiveAndDime system that is used to illustrate many of the concepts and methods in this book is an example of such a comprehensive system, Web based upon open source software. It system uses the MySQL database product and the PHP application server, and it will operate in an open source mode (Linux/Apache) or on proprietary platforms such as Microsoft Windows/IIS or a Unix operating systems. The general application framework for FiveAndDime is shown in Figure 15.10.

Figure 15.10. FiveAndDime application architecture

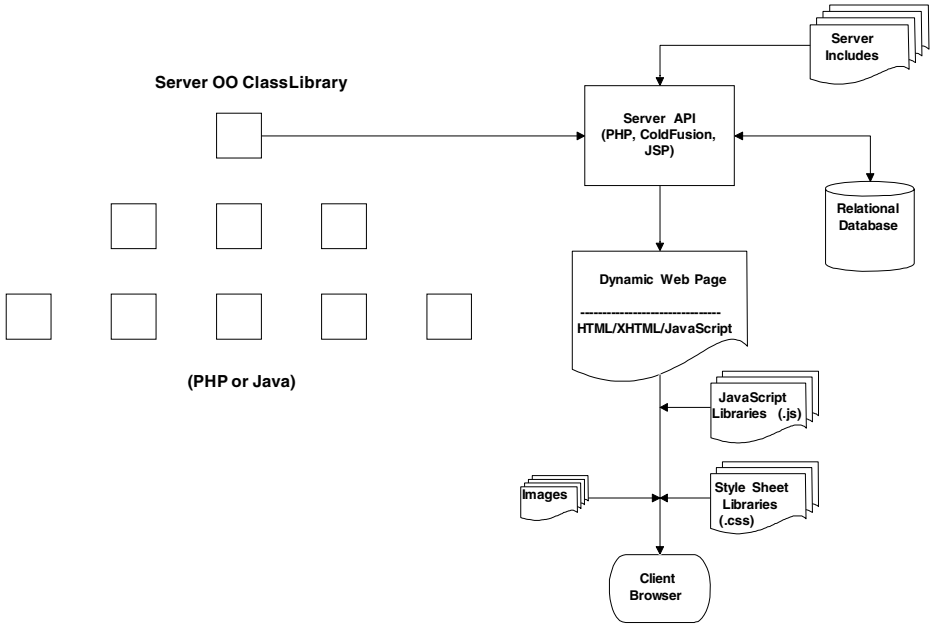


Figure 15.11. Earned value analysis superimposed on Gantt chart

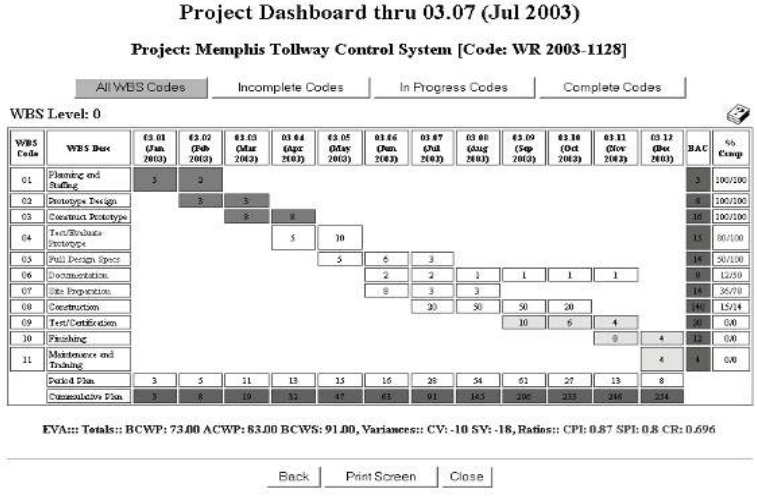


Figure 15.12. Earned value drill down

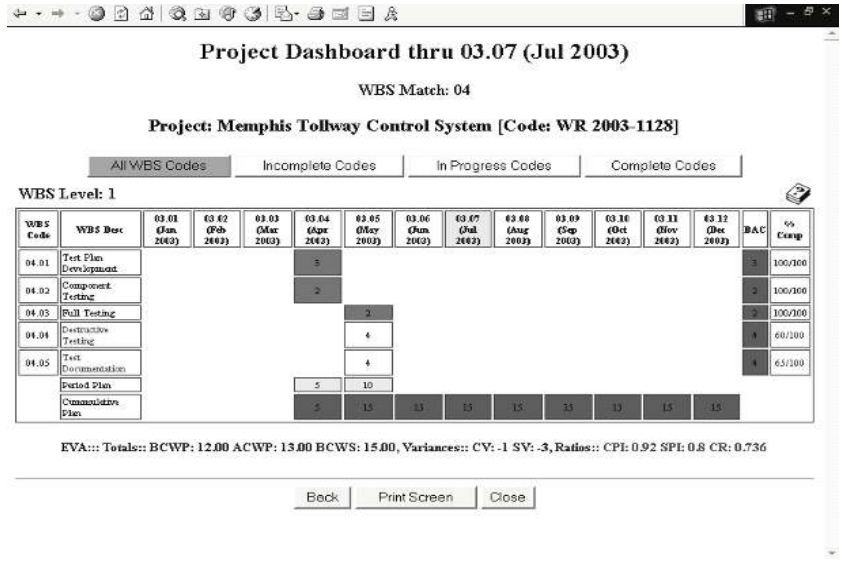


Figure 15.11 is a screen print from the FiveAndDime system of a rollup to the top level of a WBS showing progress (percentage complete), plan costs, actual costs, and earned value.

Figure 15.12 is another screen print from the system that shows the drill-down option. When one clicks on a top level WBS, code the system shows the subsidiary WBS codes (used to identify problem spots in a project). Note that the screen shots in the figure are black and white, but the actual screens are in color, to highlight WBS codes with cost and/or progress problems.

Chapter Summary

In this chapter, software for project management has been discussed. Spreadsheet models have been illustrated for a number of common project management tasks, software for specific project management functions have been identified, and general project management systems have been listed. open source software as it relates to project management was also discussed and illustrated. *However, these tools do not replace the need for a PM's complete understanding of the project management discipline, because "a fool with a tool is still a fool."*

References

Anthes, G. (2004, May 31). Sabre flies to Open Systems. *Computerworld.*

Babcock, C. (2004, March 29). Open Source, Part 2. *Information Week*.

Banerjee, P. (2001, September 18). Open to discussion. *Intelligent Enterprise*.

Dyck, T. (2003, March 26). SQL databases—Clash of the titans. *PC Magazine*.

Greenemeir, L. (2004, May 24). Going main stream. *Information Week.*

Klastorin, T. (2004). *Project management—Tools and trade-offs.* New York: Wiley.

Wheatly, M. (2004, March 1). The myths of open source. *CIO Magazine.*