



Bellwether Consulting

Bellwether Consulting Presents
Software R&D Management Series

Metrics and Reporting for Development Visibility

by **Stan Bell**
Principal Consultant

Visibility is a critical factor in successful software development management. You can't manage what you can't see. "Seeing" is about improving your subjective feel by adding objectivity. Empirical data expand understanding of the development team and the state of it's on-going activities. Continuous, systematic measurement enables management to recognize and treat the right problems, and recognize and reward real successes. Over time, continuous interaction with the data improve intuitive understanding, as well.

This paper describes a set of reports that address software development and delivery, ie, execution of the development function. The set is founded on just a few direct measures. To use these reports effectively implies other robust management activities are in place, such as estimation and risk management. Consider this guide an end-state description, from which the individual organization might take direction for improving development management practice.

As described, these reports are meant to be used internally. They support the internal management of the development organization. The materials can be directly applied to managing subcontracted development efforts, in varying degrees depending on the engagement.

This set does not address more general management needs, such as individual performance management within the development team. However, much of the raw material presented here can be applied to performance management. This is the subject of another paper in the *Bellwether Consulting presents* Software R&D Management Series.

Reporting Audience

Depending on the scale of the organization, development reporting may be addressed to different audiences. Nonetheless, most will find these constituencies appropriate. You may find your titles or positions don't match. It is not uncommon, for example, for project management to be done by development managers. Treat these labels as roles, rather than specific positions, and understand that more than one role may be filled by a single position. In large scale organizations, multiple positions may make up one of these roles.

First, where the rubber meets the road. Project managers are both producers and consumers of much of the presented reporting. Either project managers or supporting automated systems will accumulate much of the included data. Project management here is defined as the position responsible for monitoring and reporting the project's progress. This role may or may not have direct responsibility for task assignments to project staff. Likewise it may not have direct responsibility for the project outcomes. Reports for this audience answer questions around the central theme of "is the work getting done according to plan?" and "does the plan need to be changed?" and "what might prevent the work getting done as planned?"

Development management are also often producers and consumers of reported data. Development management is defined as the role responsible for personnel management of assigned project staff. Often this role will have responsibility for task assignment and for project outcomes. Frequently this role will have responsibility for a product, which is created, modified, or maintained by a series of projects. This role sometimes has responsibility for the contribution of a particular discipline, such as business analysis, engineering, architecture, or quality assurance. Questions for this audience tend to be in the range of “am I using my capacity effectively?” and “is the work being performed to standard?” and “what might divert my capacity from assigned projects?”

Product management is the role responsible for identifying market requirements. This role determines appropriate functional content, and changes to functional content, in a given product or suite. This role is typically concerned with the effect on business success of a product, rather than it's technical implementation. Product management is concerned with the available capacity to apply to a given product, ie, how much can be built, in what timeframe. Product management is also concerned the product content satisfies market requirements. In addition to those questions, product management might is also seek status information, “will the release plan be met at current pace?”

For purposes of this paper, product management is also assumed to be the communication channel for other stakeholders. These include other functions of the business such as services, sales, and support. Most important as stakeholders are interested customers.

Portfolio management is the role with responsibility for how an organization's total assets are allocated to various products or suites. Allocation among the products is usually done on the basis of return on assets or return on investment. That is, which product investments return the most gain to the overall business? Status information of concern is “are prior investments on track to produce the promised gain?” and “what capacity is coming available?” and “what additional investment opportunities are coming?” This latter question is not addressed by the reports described in this paper.

P&L owner is labeled thus for convenience, the role to whom the development function has accountability as a function of the business. In shared service organization models, actual P&L responsibility is often peer to R&D functional leadership. In any case, there is a title or position somewhere in the organization to whom R&D is accountable. Questions that characterize this audience's interest are also around execution, “are projects progressing on plan?” Most other questions derive from this one, or seek to explain variances. For example, “are projects staffed adequately?” and “are projects estimated accurately?” and “are capital assets in sufficient supply and used efficiently?” and “is the function performing within budget?”

Enterprise R&D governance is the most likely of these audiences to be missing from a smaller-scale organization. It's role is to oversee development practices across multiple development teams, usually to set policy and monitor compliance on legal, market, investment, technology, and various other subject matters. Often such a group is or oversees a methodology, best practice, or architecture clearinghouse. Theme of interest for this audience will be comparative performance of the various groups, ie “which practices produce the best results?” Policy and compliance reporting will not be addressed in this paper, but some metrics of interest for inter-team comparison will be suggested.

Reporting Frequency

There are two broad types of chronology for which visibility is needed. First is routine reporting, that

may be done weekly, monthly or quarterly. Reports tied to project milestones also fits this category.

The second broad type is event-driven. It is perhaps best understood in contrast. Routine reporting is done on the elapse of the calendar or on the occurrence of the milestone, regardless of whether the outcome is positive or negative. More importantly, it is regardless of whether any response action is required. Event-based reporting is in direct response to some *unpredicted* event that almost always requires some management response. Most of the events that can occur can be anticipated categorically, if not individually, and are often realized risks. The general categories are these:

- content change—new requirements, scope change decision, design change decision
- capacity change—addition or reduction of resources, eg medical leave
- capability change—acceleration or deceleration in velocity, increase or decrease in defect rate.

For full visibility, a robust reporting system should plan to report on these categories of events. The content of reports may be the same for event-based reporting as for routine reporting; they're just presented on a different time line.

To select routine reporting frequency, balance the effort required to prepare reports with the value of the information. When establishing a new reporting system, it's often best to err on the side of shorter frequencies. This may be counter-intuitive. Reasons are it's easier to tune to a less-frequent model than to a more-frequent once the routine is established. More-frequent spreads the work of collection and preparation in smaller portions over more time. Infrequent reporting tends to drive procrastination followed by a flurry of activity near the deadline. This style of interaction with the metrics and reports has the undesirable affect of making the reporting an extraordinary event, distinct from the conduct of the work. The affect you want is for teams to engage directly with and have sensitivity to the metrics, where they are a natural side affect of the work they describe, requiring a minimum of additional effort.

An analogy of direct engagement: if your job is to run the smelter for the production of molten steel, you won't think of the taking the temperature of the steel as extra work. You can't do your job effectively without it.

When starting up, balance the recommended high frequency with a shorter list of metrics, to avoid immediate overload and burnout. Start with just the essentials, the core metrics. In the descriptions that follow, certain reports are called out as most valuable to start with.

A General Reporting Approach

A simple way to approach reporting is to emulate standard budget reporting. This is the approach advocated here. Generally, this approach is applicable to routine reporting, but can be applied to event-based reporting as well.

Budget reporting typically follows a pattern of showing actual values for the months of the year already concluded, and forecast values for the remainder of the year. These values are compared to an established baseline, the budget. Then variances are shown, calculating by how much the actual and forecast vary from the budget. Variances are typically shown as absolute dollar amounts and as percentage differences. Significant variances are usually accompanied by an textual explanation.

Forecast values for future months are modified periodically as conditions change throughout the year.

When the actuals deviate too far from the budget to be useful, a new baseline is established. This new baseline supplements, rather than replaces the original budget. The new baseline is often a snapshot of the forecast at the time the baseline is taken, and therefore is often referred to by some vernacular in practical use, eg “the November forecast.” This new baseline may become the working standard of comparison.

This same approach can be applied to R&D reporting. A set of metrics can be treated in a manner similar to budget line items, each with an initial forecast (estimates). The accumulation of the various metrics make up the measurable elements of a plan. They can be time bound in the same way as a budget, eg monthly. Then in on-going reporting, the actual values for each metric are reported for the period, and forecasts for future period's values are revised. These new actual and forecast values can be compared to the original plan. Variances can be shown as absolute amounts in the units of the corresponding metric, and as percentage deviations from the original planned values.

As with financial reporting, the variances can be a trigger to indicate a new baseline is necessary. The threshold values that prompt a new baseline should ideally be planned in advance, but may also be identified as they arise.

In establishing a plan that includes planned values for all the metrics described below, the R&D organization is driven to consider elements that tend to improve the quality of the plan overall. Examples are estimating and forecasting defect rates, productivity, velocity, others. These become key assumptions of the plan.

Core Metrics

The reports described below are predominantly based on four core elements. These are

- size,
- effort,
- duration, and
- quality.

Each of these is measurable with different specific metrics, in different units of measure. A sufficient metrics and reporting program should minimally address each one.

Most will choose to represent effort as a time unit of work, as well as a corresponding cost of that unit, eg person-hours and average cost of a person-hour. Non-effort costs too are an important supplement to the core elements.

Duration describes the elapsed or calendar time over which the effort is applied. Since effort is never entirely continuous, it is necessary to also apply duration for clarity in reporting. Duration is often a more important constraint than effort.

Size may be measured as function points (see www.ifpug.org for a complete description), feature points, lines of code, counts of data or UI elements, or any variation on this theme. Size measure is problematic for many R&D groups, but is important for understanding development performance as

well as effective estimating and planning.

Many development teams represent size as effort. Unfortunately, they're not the same thing. A favorite metaphor is of a fence: regardless of the material of which it is built, a fence can be the same size. However, that size built of bricks vs boards vs wire vs hen's eggs will all require different levels of effort. Size of abstract software is a challenging concept.

Quality can be measured in many ways, as well. Fundamentally, quality is difficult to describe, but we know what it's not. Perhaps this explains why counts of defects or faults is a common way to measure quality. That is, quality is the absence of defects. This approach is convenient and functional for most software development purposes.

The Reports

The remaining sections describe specific reports.

Capacity and Utilization

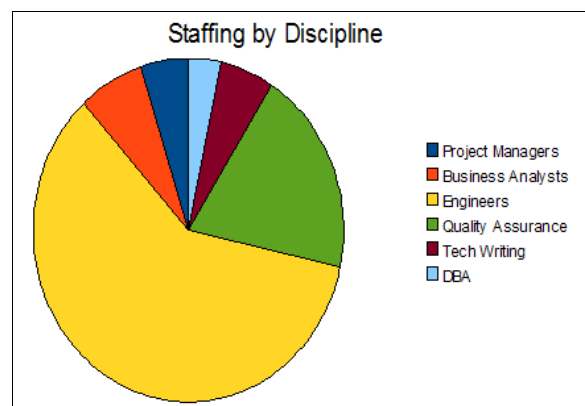
The purpose of capacity and utilization reporting is to provide a broad view of how resources are allocated among development categories, products, disciplines, and projects. The broad view provides visibility into relative consumption of resources, and enables a subjective value-for-investment assessment. It shows the total capacity available to the organization, and therefore can highlight constraints on production. The “utilization” aspect refers to the actuals, which reflect how resources were actually used rather than how planned. Actual use is derived from reported effort expended, divided by an FTE factor.

Secondly, the capacity report can be used as a working tool to negotiate resource assignments by development and project managers.

If you're just starting out, just identifying where you're using your human resources can be enlightening. You don't have to have high precision. You will get value from seeing where individual staff are working on multiple projects. As you will from seeing how your resource is allocated to different efforts in aggregate, such as divided between maintenance and new product development.

Capacity and utilization should be reviewed at least monthly.

An example of a capacity report follows. Portions of the report have been cut out for brevity. This example shows actual and forecast for a calendar year, with capacity by discipline shown as “total employment.” Utilization shows sum of actual staff used, built up from the detail projects illustrated below. A column at the far right shows staff in each discipline as a percent of total capacity, labeled “allocation.”



This report shows utilization variances as % under- or over-utilization in rows that describe types of staffing. Under-utilization is shown in red, over-utilization in black. Total utilization and production utilization are shown as straight percentage, ie 100% means capacity fully utilized.

“Production” utilization refers to engineering staff. It does not mean other staff are not productive, of course. Engineering capacity is often the limiting factor in planning how much software can be produced, and is a useful partial view.

Capacity & Utilization Report											
Month	Actual	February	March	April	Forecast			November	December	Person-month:Allocation	
Month	January	February	March	April	May	June	July	November	December	Person-month	Allocation
Total Employment											
Project Managers		3	3	3	3	3	2.5	3	3	35	5.0%
Business Analysts		4	4	4	4	4	4	4	4	48	6.8%
Engineers		34	34	35	34	34	35	36	36	422	59.8%
Quality Assurance		11	11	11	11	11	11	12	12	138	19.5%
Tech Writing		3	4	4	4	3	3	3	3	39	5.5%
DBA		2	2	2	2	2	2	2	2	24	3.4%
Total Staffing		57	58	59	58	57	57.5	60	60	706	100.0%
Production Capacity		34	34	35	34	34	35	36	36	422	59.8%
Utilization											
Project Managers		3	3	3	3	3	3	2	1	32.5	
Business Analysts		4	4	4	4	4	4	2.5	1	43	
Engineers		34	34	34	34	34	34	23	8	366	
Quality Assurance		12	12	12	12	12	12	8	3	129	
Tech Writing		2	2	2	2	2	2	1.5	0.5	21.75	
DBA		2	2	2	2	2	2	1.75	0.25	22	
Total Utilization		57	57	57	57	57	57	38.75	13.75	614.25	
Production Utilization		34	34	34	34	34	34	23	8	366	
Utilization, % over/under Capacity											
Project Managers						20%				-7%	
Business Analysts											
Engineers				-3%		-3%		-36%	-78%	-13%	
Quality Assurance		9%	9%	9%	9%	9%	9%			-7%	
Tech Writing		-33%	-50%	-50%	-50%	-33%	-33%	-50%	-83%	-44%	
DBA											
Total Utilization, % Capacity		100%	98%	97%	98%	100%	99%	65%	23%	87%	
Production Utilization, % Cap		100%	100%	97%	100%	100%	97%	64%	22%	87%	
Project Utilization Detail											
Product A											
Project A-1											
Maintenance											
Project Managers		0.5	0.5	0.5	0.5	0.5	0.5			4.5	
Business Analysts		0.5	0.5	0.5	0.5	0.5	0.5			4.5	
Engineers		5	5	5	5	5	5			45	
Quality Assurance		2	2	2	2	2	2			18	
Tech Writing		0.25	0.25	0.25	0.25	0.25	0.25			2.25	
DBA		0	0	0	0	0	0			0	
Total FTE		8.25	8.25	8.25	8.25	8.25	8.25	0	0	74.25	
Production FTE		5	5	5	5	5	5	0	0	45	
Project A-2											
Enhancement											
Project Managers		1	1	1	1	1	1			11	
Business Analysts		1.5	1.5	1.5	1.5	1.5	1.5			16.5	
Engineers		15	15	15	15	15	15			165	
Quality Assurance		5	5	5	5	5	5			55	
Tech Writing		1	1	1	1	1	1			11	
DBA		1.5	1.5	1.5	1.5	1.5	1.5			16.5	
Total FTE		25	25	25	25	25	25	15	15	255	
Production FTE		15	15	15	15	15	15	15	15	165	

The pie-chart illustration is taken from the allocation column in the report example.

Project Status Reporting

The purpose of project status reporting is to show how actual execution is corresponding to plan. This provides opportunity for adjustment. Remedial action may be necessary if actuals are behind plan, and re-planning may be necessary if actuals are exceeding plan.

Relative values of metrics can expose areas for further inspection, to uncover problems or opportunities. Over very little experience, managers will develop an intuitive sense of how a project progresses in terms of each of these metrics, and will understand what “normal” numbers are. This

makes any discrepancy an opportunity for inspection. Closer attention to anomalies leads to quicker problem detection and quicker cure, which ultimately leads to successful project outcomes.

Effort, cost, size, duration, and quality should be reported.

Effort is most often measured as person-hours, but may also be a larger unit such as person-days. Smaller-grained measures bring higher precision, but not necessarily greater accuracy. Staff generally will not exercise much care in reporting accurately to the degree implied by the measure. And the reporting itself is not productive time, so it's preferable to minimize the burden. Larger-grained measures are generally more appropriate for larger projects.

It is also useful to show the actual number of persons employed on the project. Access to adequate resource is often a determinant of project success. Illustrating actual staffing will sometimes explain variances in effort.

Full-time equivalents (FTE) are a reasonable unit in which to report staffing. However, take care that an FTE report does not mask excessive part-time contribution. That is, if it requires 5 people to make up one full-time equivalent, the project will not receive equivalent value as if one person made up the FTE. Twenty percent of five persons' time is not of the same value as 100% of a single person's.

Cost typically has two components. First is directly derivative of effort; person-hours * cost per hour yields labor cost. Managers can choose to include or exclude overhead, allocations, non-direct labor expenses in the cost per hour factor, as long as the factor is applied the same way throughout the reporting series.

Second cost is everything non-labor-related, and may include travel expenses, equipment expenses, facilities, software tools, staff training, use of shared resources such as data centers, and many other factors. Again, consistency is important throughout the reporting series. Non-labor costs should not be ignored as they are often a significant fraction of project costs.

If components of projects are outsourced as fixed-price deliverables, payment events according to contracted terms should be included as component costs of the project. A third category is recommended.

Size is one measure many development shops avoid, or equate with effort. Size should be measured and reported independently. Function points are recommended. Second-choice measures are KLOC or thousands of lines of code, or feature points. Both will require consistent definition. Feature points can be derived from number of elements on screens, number of input or output elements to services or to persistent stores, etc. Feature points are harder to define unambiguously but usually easier to count and for non-technical stakeholders to understand. KLOC is a traditional measure, for which some project history data is available, but is difficult to predict and is a poor predictor itself of project progress.

Managers may also choose to count artifacts. These may be source files, dll's, documents, objects, services, use cases, specifications, or whatever is readily accessible. This is a very coarse measure, but still much better than no size measure at all.

Duration is, of course, measured by elapsed calendar time. Project plans typically lay out tasks on a

schedule. Tasks or sets of tasks typically have an endpoint deliverable, which indicates the task is complete. Given this source material, a time line can be established that show days (weeks, months) ahead or behind as each task is completed. A simple new forecast for the project duration can be made by adding the net of late and early dates to the end of the project. These can be shown numerically but graphical depiction is usually more effective.

It is often very useful to compare the planned distribution of labor over calendar time with the actual use. This can be a powerful leading indicator of project schedule success. Given the project schedule,

Project Status Report										Project to Date		Forecast to Complete	
Metric	Actual			Forecast					Variance	% Variance	Variance	% Variance	
	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep					
Staffing, FTE	5.5	9.5	10	10	11	11	10.75	10.25	-2.7	-10%	-1.8	-2%	
Budget	7.25	10.2	10.2	10.2	10.2	10.75	10.75	10.25					
Variance	-1.8	-0.7	-0.2	-0.2	0.8	0.3	0.0	0.0					
% Variance	-24%	-7%	-2%	-2%	8%	2%	0%	0%					
Cost, \$000	55.4	91.4	83.5	82	92	86	86	86	0.1	0%	13.5	2%	
Budget	64	83.1	83.1	83.1	83.1	86.4	86.4	79.6					
Variance	8.6	-8.3	-0.4	1.1	-8.9	0.4	0.4	-6.4					
% Variance	13%	-10%	-0%	1%	-11%	0%	0%	-8%					
Effort, person-hours	755	1596	1623	1680	1760	1936	1806	1640	-474	-11%	-336	-3%	
Plan	1102	1714	1632	1714	1632	1892	1806	1640					
Variance	-347	-118	-9	-34	128	44	0	0					
% Variance	-31%	-7%	-1%	-2%	8%	2%	0%	0%					
Size, cum function points	114	278	545	920	1130	1430	1570	1800	-35	-6%	0	0%	
Plan	170	260	580	940	1150	1440	1570	1800					
Variance	-56	18	-35	-20	-20	-10	0	0					
% Variance	-33%	7%	-6%	-2%	-2%	-1%	0%	0%					
Size, artifacts completed	7	14	28	46	57	72	78	97	-19	-40%	0	0%	
Plan	13	29	47	58	72	79	90	97					
Variance	-6	-15	-19	-12	-15	-7	-12	0					
% Variance	-46%	-52%	-40%	-21%	-21%	-9%	-13%	0%					
Quality, discovered defects	57	139	218	276	226	286	157	72	-33	-7%	0	0%	
Plan	85	130	232	282	230	288	157	72					
Variance	28	-9	14	6	4	2	0	0					
% Variance	33%	-7%	6%	2%	2%	1%	0%	0%					
Quality, remaining defects	6	28	45	35	25	20	16	5	-22	-96%	2	29%	
Plan	8	13	23	28	23	29	16	7					
Variance	2	-15	-22	-7	-2	9	0	2					
% Variance	25%	-115%	-96%	-25%	-9%	31%	0%	29%					
Quality, % test executed	100	74	63	70	75	80	90	100	63	-21%	0	0%	
Plan	100	90	80	80	80	90	90	100					
Variance	0	-16	-17	-10	-5	-10	0	0					
% Variance	0%	-18%	-21%	-13%	-6%	-11%	0%	0%					
Duration, forecast calendar days	248	251	251	252	250	249	249	249	-8	-3%	-6	-2%	
Plan	243	243	243	243	243	243	243	243					
Variance	-5	-8	-8	-9	-7	-6	-6	-6					
% Variance	-2%	-3%	-3%	-4%	-3%	-2%	-2%	-2%					

the number of hours that should be consumed each day (week, month) can be computed. Comparing these numbers with the actuals as they occur can suggest the project is ahead or behind. If this indication doesn't match the size result, then estimates are inaccurate. Significant deviation may trigger a re-planning event.

Quality has to be tracked to determine if the work progress is actually complete to a deliverable standard. Recommended metrics are numbers and percentages of test cases written and passed, numbers of test cases executed, and executed cases which have passed. Test cases executed, both planned and actual, will exceed number written early on, as tests are executed multiply but written

once. Therefore, for clarity it's useful to show percentage passed as a fraction of total cases as well as as a fraction of total executed. This prevents misunderstanding resulting from repetitive execution of passed cases.

Above is an example of project status reporting. The example shows February through September, an eight-month project. Generally, it is appropriate for project or development managers to follow these metrics on a weekly or bi-weekly basis, while reporting to stakeholders outside the project on a monthly basis.

#	Risk Item	Probability %	Impact 1-5 scale	Score C/D	Base Action	Owner	Mitigation Plan to prevent	Cost Basis	Contingency Plan if realized	Cost Estimate	Weighted Cost Risk C/K
5	Inadequate quality Staff not freed from prior projects, not available	30%	4	1.2	re-work	John	plan/estimate with historical quality info, adjust for project factors; change methodology to test early on smaller scale; engage QA in requirements and design	\$60,000/month market cost of delay; addl project expense of addl labor est \$60,000/month opp cost of late to market; market share % loss	delay project for re-work	\$120,000.00	\$36,000.00
2		20%	5	1	delay	John	monitor predecessor projects, assist to complete if feasible	\$60,000/month market cost of delay; addl project expense of addl labor	delay project	\$120,000.00	\$24,000.00
					alternate staffing			staff morale	hire contract staff	\$288,000.00	\$57,600.00
4	Productivity too low	30%	3	0.9	re-estimate	John	plan/estimate with historical productivity numbers, adjust for project differences	\$60,000/month market cost of delay; addl project expense of addl labor	delay project at new pace	\$120,000.00	\$36,000.00
1	Cannot acquire customer-partner	10%	5	0.5	requirements	Mary	early solicitation; engage sales & services teams in identifying candidates		hire industry consultants	\$75,000.00	\$7,500.00
6	Medical Leave losses	25%	2	0.5	absorb loss	John	Poll staff for potential; back-fill with alternate staff on deck	staff morale	overtime if low loss	no direct cost	0
					alternate staffing			70-120/hour for contract staff	if higher loss time	\$80,000.00	\$20,000.00
7	Regulatory change to algorithm	40%	1	0.4	adopt change	Mary	monitor federal register, legislation; weekly contact with personal sources	\$60,000/month market cost of delay; addl project expense of addl labor	add effort to project	\$20,000.00	\$8,000.00
8	Late requirements additions	20%	2	0.4		Mary	validate requirements with stakeholders frequently; designate specific individuals, review req's biweekly, validate with customer partner biweekly	\$60,000/month market cost of delay; addl project expense of addl labor	delay project	\$60,000.00	\$12,000.00
3	Component X not available/acceptable to buy	5%	4	0.2	build instead	Tom	addl research, check references, forums, run downloadable trial versions		build in-house	\$146,000.00	\$7,300.00
										\$208,400.00	

Each measure shows actual/forecast, plan or budget, and variance of actual/forecast to plan/budget in units of the metric, and then in % variance. Negative variances are shown in red, positive in black. Large variances are in boldface.

In use, a threshold should be set at a level appropriate to each metric. When beyond the threshold, variances would be highlighted (eg, boldface) and explained. A similar practice of threshold-setting is a very effective way to identify when re-planning is necessary on a project. More on this point in another paper in the series.

Risk analysis for each project should be reported routinely. While not metrics per se, the risk assessment performed periodically improves project success. The set of risk elements will include the baseline risk assessment, risks realized, risks deprecated, and new risks identified. Risks realized are those which have come to pass, and should indicate what response has been triggered. (If the risk did not warrant a response plan, it probably should be excluded from the risk analysis.) Risks may be deprecated because mitigation has reduced their significance, they may have been resolved by some external change of circumstance, they're no longer applicable. Newly identified risks should enjoy the same attention that risk analysis originally applied, and those deemed to merit inclusion in the mitigation or contingency plans should be called out.

An example risk analysis instrument is shown here. In this example, risks are sorted by the composite score, which is probability of occurrence * impact

to the project, on a scale of 1 to 5. This example shows a typical initial state. Monthly reporting should show changes, including new risks, deprecated risks, and changes of probability or impact of existing risks. Similarly, cost or feasibility of mitigation or contingency may change, causing a change in the overall risk picture of the project.

This example shows every listed risk with a contingency plan and cost estimate. For the most part, contingency plans are not warranted for low-impact or low-probability risks. Mitigation actions should be scaled likewise, to the impact and probability.

The sum of weighted risk costs is not a “real” number in itself. It can be viewed as an overall indicator of project risk when compared to total project costs. That is, if the weight cost sum is 10% of project cost, that represents relatively low risk overall. If 90%, substantial re-planning or heavy emphasis on mitigation is in order. The magnitude of this percentage might be used to influence estimation, if using a best-case/likely-case/worst-case estimate weighting.

Group & Portfolio Performance

Derivative metrics show the on-going performance of the development team over time, and over multiple projects. Development managers can develop solid macro estimation capability with accumulated history of project characteristics and team performance. Memory does not serve, but systematic records of performance will.

In a larger enterprise, these metrics can be used to compare performance among multiple development teams, and for multiple product lines or technologies. Variations in performance among teams can identify good practices to be propagated, or less useful practices to be remediated. The source of variation may be external conditions as much as team capability, such as intense regulation or complex business logic. Nonetheless, it's better to be in a position to ask informed questions than to be in the dark.

An example cross-project metrics illustration follows. In this example, especially positive values are highlighted in green, and negative values in red. The report is divided into two illustrations for legibility.

Projects	Descriptive						Completion		Perf
	Type	Size	Size	Planned Duration	Effort	Quality	Size	Duration	Qt
		Total	New	Total	Total	Total	Remaining	Remaining	Def
		Function Points	Function Points	Months	Person-Months	Defects	Function Points	Months	Defec
Group A									
Project A-1	New Product	14,700	11,550	18	702	656	3,356	7	
Project A-2	Enhancement	1,800	1,660	8	114	68	425	2	
Project A-3	Maintenance	2,112	248	4	32	10	139	2	
Group B									
Project B-1	Enhancement	6,450	6,450	11	418	315	2,300	4	
Project B-2	Enhancement	796	710	6	60	17	353	3	
Group C									
Project C-1	Enhancement	3,318	2,878	6	198	179	291	2	
Project C-2	New Product	16,250	16,250	19	1,710	618	10,634	11	
Project C-3	Maintenance	951	243	3	12	3	146	1	
Overall		46,377	39,989		3,246	1,866	17,644		

Metrics included are categorized as descriptive, as indicating degree of completion, and as technical and financial performance.

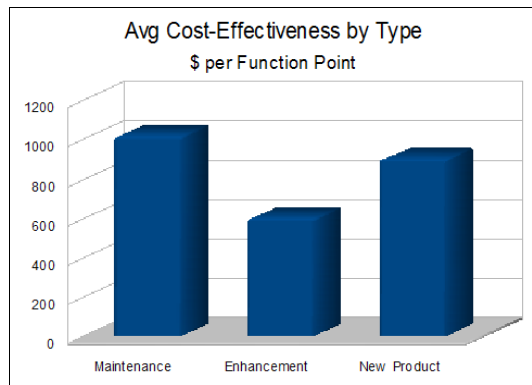
Descriptive metrics are total size and net new size, where the difference represents re-use. Duration is shown in months. Effort is shown in

person-months. Quality is shown as total defects discovered. Type categorizes the projects by the nature of their work on the affected products.

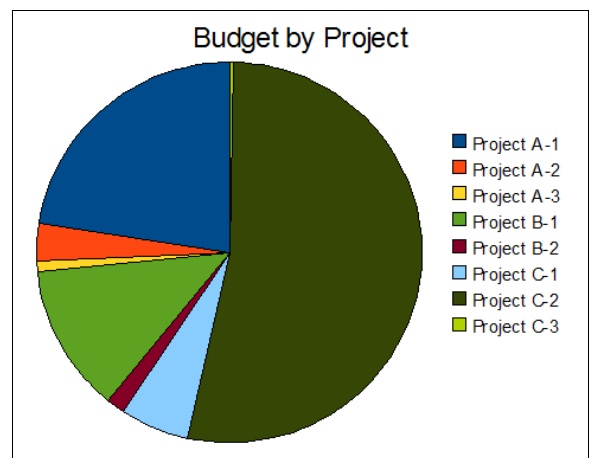
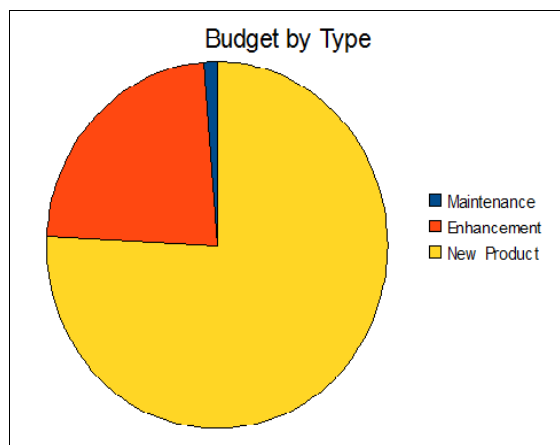
Completion is expressed as remaining size, in this case function points not yet completed, and remaining duration.

Projects	Descr Type	n g	Performance				Financial					
			Quality	Velocity Size/ Duration	Productivity Size/Effort	Cost- Effectiveness Size/Cost	Cost	Cost Spend to date	Budget Variance %/to date	Remaining Spend Forecast	Cost at Completion Forecast	Budget Variance at Completion Forecast
			Defects/Size	NewFP/ Person/ Month	Hours/FP	\$/FP	Total \$000	\$000	pos/neg %	\$000	\$000	\$000
Group A												
Project A-1	New	7	0.080	19.1	8.4	532.29	7,137	4,362	-4.00%	1,786	6,148	14%
Project A-2	Enhanc	2	0.055	14.5	11.0	597.70	985	738	6.50%	254	992	-1%
Project A-3	Mainte	2	0.094	6.8	23.5	1,348.04	293	147	-1.10%	188	334	-14%
Group B												
Project B-1	Enhanc	4	0.076	15.6	10.3	608.97	3,971	2,527	-0.40%	1,401	3,928	1%
Project B-2	Enhanc	3	0.048	11.9	13.4	707.28	505	253	-2.12%	250	502	1%
Group C												
Project C-1	Eni	2	0.069	19.6	8.2	467.69	1,815	1,210	4.34%	136	1,346	26%
Project C-2	New	11	0.110	7.8	20.5	1,260.68	16,815	7,080	-7.40%	13,406	20,486	-22%
Project C-3	Maini	1	0.036	12.1	13.2	674.93	98	65	-1.04%	99	164	-67%
Overall			0.071	13.4	13.6	774.70	31,619	16,381	-0.65%	17,519	33,901	-7%

Performance metrics are all normalized to allow for easy comparison across disparate projects. Quality is expressed as defects per unit size. Velocity is size per unit duration, in this case new function points per person per month. Productivity is size per unit effort, or the inverse, effort per unit size. The specific metric here is hours per function point, a moderately available metric for benchmarking. Cost-effectiveness is size per unit cost inverted, or dollars per function point.



Progress against plan is visible across a product portfolio with this collection of metrics. These metrics enable a big-picture view of development investment and results, as well. These chart examples are made possible by the capture of the core metrics. These should inspire addition of other useful elements, appropriate to your circumstances.



Schedule

Project managers should work with metrics that contribute to these reports more or less continuously. Other managers will typically have less immediate need. At minimum, the reports should be reviewed according to this schedule.

	Project	Development	Product	P&L	Portfolio	R&D Governance
Capacity & Utilization	Bi-Weekly	Bi-Weekly	Monthly	Monthly	Quarterly (Roll-up)	
Project Status	Weekly	Weekly	Monthly	Monthly		
Risk Analysis	Bi-Weekly	Bi-Weekly	Monthly	Monthly		
Group & Portfolio		Monthly		Quarterly	Quarterly	Quarterly

Bear in mind event-triggered reporting is no less important, even though it does not follow a predictable schedule. Recommended trigger events are shown below; a trigger is a change in the identified subject. The percentages are a suggested threshold, defining significance of the change for the corresponding audience.

	Project 5%	Development 5%	Product 5%	P&L 10%	Portfolio 25%	R&D Governance 10%
Capacity & Utilization	Content Capacity	Content Capacity	Content Capacity		Capacity	
Project Status	Content Capacity	Content Capacity	Content Capacity	Content Capacity		
Risk Analysis	Content Capacity	Content Capacity	Content Capacity	Content Capacity		
	Risk Realization	Capability Risk Realization	Risk Realization	Risk Realization		
Group & Portfolio				Capability	Content Capacity Capability	Capability

Summary

Following the recommendations in this paper will accomplish visibility into the development organization. The minimum set of measures are size, effort, duration, and quality. Optimally, include cost as a partial function of effort and duration, with other contributing factors.

These materials are equally applicable to traditional methodologies, or to modern agile methods. Simply adjust the specific metric to a measure appropriate to your process. With short-iteration methods, it may be appropriate to adjust the reporting frequency to match iteration frequency.

The reader is encouraged to develop a locally-suitable set from these.

Contact the Author

For detailed guidance in creating an R&D visibility system based on these recommendations, contact Bellwether Consulting at info@bellwethermgt.com or the author directly at stan.bell@bellwethermgt.com.



Software Development from Art to Science