Weighted Incident and Project Metrics Using Visual Intercept

Three of the new Visual Intercept report formats released this month calculate incident metrics based on incident weight values. This month's Tech Tip describes how each of these reports was created to provide a particular type of metric and how you can use these reports, or reports similar to them, to generate your own metrics for incidents and projects.

We will start with a brief description of weight values, how they are set and why they are important to the generation of useful project statistics. We will then look at each of the incident metric reports included in the Visual Intercept reporting supplement. Each of these reports utilizes a different formula to calculate the total weight value for each incident, and each formula is designed to generate a particular type of metric that would be useful during a different phase of the product development cycle.

We will also explore some ways you can leverage Visual Intercept's ability to generate reports from query results to apply these metrics to specific kinds of cases. Finally, we will take a look at how you can generate your own incident metrics as well as some alternative ways you can use weight value metrics to increase efficiency and speed development processes at your organization.

Each of the reports included in this supplement is designed for a particular type of incident metrics that we have found valuable at Elsinore. Thus, we submit these reports as useful starting points – not as definitive statements of best practice. If you have metric formula that you believe would be useful in this context, please submit them to: support@elsitech.com.

What are Weight Values?

In the Visual Intercept Administrator for both Visual Intercept Enterprise Web and Desktop, you can associate a numerical value for each of the values you set for any of the drop-down lists in the application. Weight values are set in the same dialog as the values for each of the document parameters. The weight value is stored as a float. Therefore, you can set weight values to be either whole or real numbers. For the purposes of incident metrics, we will be concerned with the weight values for incident status, severity, priority and category.

The Fundamentals of this Approach to Metrics

How can weight values be used for metrics? Let’s take a very simple case: the incident priority. As the term "priority" is typically interpreted, the higher the priority of an incident, the less option there is to not resolve it. Low priority incidents can be resolved as existing resources are become available, high priority incidents don't provide the same option, and medium priority incidents fall somewhere in between. Thus, the more high priority incidents, the more load there is on existing resources and the more likely the overall project is at risk if no intervention is taken.

For the purposes of assessing risk, resource load, and other properties of an ongoing project, an organization could simply create reports that count the numbers of incidents of different types. Reports that count "open" versus "closed" incidents and compare those totals to totals made on regular intervals in the past or reports that total incidents by priority are common examples of this type of metric.

Metrics that involve totals based on attribute types certainly have their place. Knowing that there are more "high" priority incidents then there were last month or that the "open" versus "closed" ratio is going in the wrong direction is important. However, metrics that simply rely on totals of attribute values can hide or make it difficult to understand certain features are important to what the incident data has to say about the current state of a project or the significance of an individual incident. Metrics based on attributes and totals can be less than ideal in these ways:

1. **The attribute values are always qualitative in nature.** Without some quantitative definition of each attribute that can be incorporated into the metrics, the interpretation of results is more likely to be subjective or to vary between samples.

2. **Understanding how one set of totals compares to another for a single attribute can be difficult.** What does it mean, for example, if one month there are 25 "high" priority incidents, 30 "medium", and 15 "low" and the next month there are, respectively, 15, 50, and 20? Which month represented a heavier load? Did the decrease in high priority incidents compensate for the increase in low and medium priority incidents?
3. **Understanding how the relative impact of an incident is a factor of all of its combined attributes can be difficult.** For example, measuring totals of open, high priority incidents from one month to the next does not make it easy to understand how the category or severity of those incidents is relevant to understanding their relative impact on the project.

Using weight values for incident metrics does not provide a definitive solution to all of these shortcomings. However, it does help in each of the three cases identified above. By using weighted values for incident metrics you can:

1. Easily establish a single point of reference for the relative significance of each attribute value for the entire organization.
2. More easily understand how one sample distribution of a particular type of incident attribute has significance relative to another.
3. Provide calculations that allow multiple types of incident attributes, e.g., status, priority, severity and category, to be factored into a single function that allows multiple types of incident attributes to be used to assess the impact of individual incidents and groups of incidents.

We are not claiming that the use of weighted metrics is a perfect solution. However, we hope to show how you can use Visual Intercept's weight values and flexible reporting to produce metrics that mitigate many of the shortcomings of metrics by attribute value and add another dimension to your understanding of the meaning of your incident data.

Calculating the Function of an Incident and the Function of a Project

As the basis for the metrics used in these reports, we incorporated the idea of the function of an incident: f(i). The function of an incident returns a numerical value based on a formula that is embedded in each report. So, based on the weight values set for the status, severity, priority and category, the function of an incident is determined by the incident attributes included in the calculation and the math that is used to generate a single value based on the weight value of each of the attributes. In these particular reports, the value returned by f(i) for each incident is displayed to the right of each incident included in the report. Thus, f(i) is simply a formula that calculates a value for each incident included in the report where the formula is designed to calculate particular incident weight values for particular kinds of metrics.

For these reports, we also incorporated the idea of a function of a project: f(p). The f(p) is simply the sum of the f(i) for a given metric report, i.e., f(p) = f(i)1 + f(i)2 + f(i)3 … . The f(p) is displayed the lower right hand corner of each report. We named this the function of a project since Visual Intercept is a project-oriented incident management system and, in its most abstract sense, a project can be understood as simply an aggregate of incidents. Thus, the f(p) can be used to sum incident weight values of all the incidents in a Visual Intercept Project. However, by leveraging Visual Intercept's ability to run a report on any query result set, the f(p) is infinitely flexible in its application.

The Incident Status as a Special Case

One other feature that is common to the f(i) for all reports included in this reporting supplement is the treatment of the incident status. We recognize that there is a particular application for which the incident status is overloaded. The incident status can be used as a weighted value where the weight decreases as the incident status indicates that that the incident is closer to final resolution. For example, the status of "closed" would have a weight value of 0 associated with it which would, if the total weight was calculated as a product, reduce the weight for an incident to 0. A status of "QA" might have a weight value of .40.

On the other hand, the weight value for incident status is sometimes used as a rough metric for percent complete. In these cases, the incident status weight value would increase as the incident moved toward closure to indicate a higher percentage of completion. Being in a status of "QA" might be judged to indicate an average of 60% to completion and be reflected by a weight value of .60 associated with it. In fact, if your organization does interpret the incident status as an indication of percent complete or percent work complete, Visual Intercept's Microsoft Project integration can automatically update the percent complete for tasks based on the status value of related incidents.

So, as a way to suggest how these to somewhat conflicting goals can be managed using the same value, we designed these reports with the idea that the weight value associated with status would be set as if it were being used for a percent complete value and that those values would be understood as inversely proportional to the interpretation of the weight value for the purpose of weighted metrics. Thus, in each of the reports that use the incident status as a value in the f(i), the status value was calculated as: 1 - (status weight value). This would allow you to have the weight value for the incident status set in the database for the purposes of percent complete, e.g., a status of "QA" would be set for a percent complete value of .6, i.e., 60%, and also use the same value as the weight, i.e., 1 - .6 = .4. Thus, if you wanted the incident status of "build" to have a weight value of .25, you would set the value in Visual Intercept at .75. Setting and interpreting the weight values in this way is simply an artifact of how these particular reports were designed. There is nothing that prevents you from designing a report that does not interpret the incident status as percent complete.

We do not necessarily recommend that the incident status be used as a guide for percent complete. We included this as a part of the reports because this is a way that some organizations interpret the incident status and we wanted to demonstrate a "clean" way in which this could be reconciled with the use of the incident status as a weight value. The objective was to demonstrate flexibility of the concept of weighted metrics when a particular attribute is overloaded.

How We Set the Weight Values

The reports included in this supplement were designed around weight values that were set as real numbers, i.e., numbers between 1 and 0. This allowed us to easily manipulate values such as the incident status to accommodate a dual use such as weight and percent complete as described above. We also believe that calculating weight values as real numbers is more convenient for most purposes. While you could also use these reports with weight values set as whole number, you would definitely want to change the reports so that they do not recalculate the weight for status as 1 minus the weight value set for each status value.

Once again, the specifics of these metric reports are designed as suggestions rather than as mandates. *See the Appendix to this Tech Tip for sample weight values we inserted into the standard Visual Intercept sample data.* Looking at these values can provide you with some additional insight into how we intend the reports to work. These values are similar to the values we set in our production database at Elsinore. If you look at the weight values, you will see that we employed the following principles when setting the values:

1. We set the weight value for the status to approach zero as the status indicated that the incident was moving toward closure. Thus, as the incident moved through its life-cycle the status value would reduce the total incident weight value as calculated in f(i);
2. Status values indicating whether an incident was closed or would not be resolved were set to 0. These were status values such as "closed," "not reproducible," "rejected,", "suspended" and "duplicate". By setting the value to 0, the status would always take the total weight of the incident to 0 if the f(i) included the status weight as a multiplier. The rationale for this is, of course, the idea that closed incidents or other incidents the organization chooses not to address represent no resource load and should be dropped from the f(p);
3. Incident attributes indicating that some aspect of the incident was undetermined or unknown were given high weight values. For example, a category or severity value of "unknown" was given a weight value of 1. The rationale in this case is three-fold:

    1. Incidents that have indeterminate attributes have them because the incidents are not fully understood. Poorly understood issues are often the most costly to resolve;
    2. Incidents that have indeterminate attributes have probably not been review or reviewed properly. Thus, there are some additional steps to be taken before the issue can be addressed properly;
    3. If you are in a position of ignorance, always assume the worst-case scenario. If the severity is unknown, assume it is a very severe issue until you have it fully

understood.

The Specific Reports Included in this Reporting Supplement

Each of the reports included in this supplement is designed for a particular type of incident metrics we have found valuable at Elsinore. Thus, we included these reports as useful starting points - not as definitive statements of best practice. There are an infinite number of ways to define the f(i) and f(p). So, you may want to think about ways you could modify these reports to suit your own best practices.

Report for Incident Submission Metrics

The Report for Incident Submission Metrics takes incidents and generates the f(i) as a product of the status, severity, priority and category weight values, i.e., f(i) = (status * priority * severity * category). This metric calculation was designed to be used during those phases of the product development cycle where you want to take a particular set of incidents--for example, all the incident being logged against a particular product--and generate a report that would allow you measure the total weight value for those incidents.

This report would be most useful for measuring the weight of incidents coming into the system prior to having them prioritized and scheduled for a release. For example, if you were running this report for incidents being submitted against a product in a maintenance phase, a spike in the total weight value calculated by f(p) might indicate the need to start planning for a product service release. This early warning would help you keep the scope of service releases within the capabilities of your team of maintenance programmers, and allow you to allocate resources more proactively.

Report for Incident Prioritization Metrics

The Report for Incident Prioritization Metrics generates the f(i) based as the product of the severity and category, i.e., f(i) = (severity * category). Thus, this report is designed for setting the priority of individual incidents based on their severity and category. The report sorts the incidents by weight value in descending order so, for example, based on the weight values you set for severity and category values, the report would calculate a higher weight value for an incident with a category of "defect" and severity of "application crash" than it would for an incident that was a "change request" with a severity of "annoying". Incidents at the top of the list would tend to merit a higher priority than those at the bottom.

This report would be most useful during the phase of product development when your team is reviewing and setting the priority of incidents to be scheduled for a particular release. By ordering the incidents by weight, you can more easily and efficiently review and set priorities based on the kinds of incidents that have been logged.

Report for Incident Throughput Metrics

The Report for Incident Throughput Metrics generates the f(i) based as the product of the status and priority, i.e., f(i) = (status * priority). This report is designed for judging the relative progress of the team at closing incidents with set priorities that are actively being worked.

This report would be most useful for generating a f(p) weight value for all of the incidents that are scheduled for a particular release. As the team worked the incidents and changed the incident status to indicate work completed for each incident, the f(p) would return a ever decreasing value as the status of each incident approached 0.

By including priority in the f(i) and setting a higher weight value for higher priority values, the resolution of high priority incidents is imparted with a disproportionately large effect on the f(p). This is important since it allows the metric to account for the fact the resolution of high priority incidents is more relevant to the success of the project than lower priority ones. Simply measuring the closure rate would obscure the fact that the incidents being closed might not be as relevant to the success of the project as one might wish.

Using the Visual Intercept's Method of Linking Queries to Reports to Refine Your f(p)

As you can see for reports such as the Report for Incident Submission Metrics and the Report for Incident Throughput Metrics, the result you are most interested in is the value returned by the f(p), and that value is simply the sum of weight values of each incident included in the report. Thus, you are going to get the most accurate and useful metrics if you can limit the incidents included in the f (p) calculation to just those incidents that are suited to the type of metric report you are running. Because Visual Intercept provides the capability to run reports on the result of any query you can create, you can see that Visual Intercept's design makes the f(p) infinitely flexible.

For example, in Elsinore's production database we use incident-to-incident relationships in addition to the Release field to track which incidents are scheduled for a particular release. Thus, I built the following query in the Query Builder that allows me to easily retrieve only those incidents that have been scheduled for the version 4.0 release by virtue of their relationship to a parent incident:

Incident.IncidentID IN (SELECT IncidentID FROM Incident_Incidents WHERE RelatedIncidentID = 4354)

Whenever I run the query, I can always run a metric report on the same results. As development gets underway, I will apply the Report for Incident Throughput Metrics to this particular set of incidents using this query. I could also achieve the same effect by running a more simple query against the Release field for the value 4.0.

So, if you understand the flexibility of the Visual Intercept Query Builder and the Search tool and how they are linked to report generation, you can begin to see how Visual Intercept allows you to generate focused metrics no matter how you choose to track your incidents. These metrics can be applied to all the incidents in the database, all the way down to the incidents assigned to an individual team member.

Modifying the Reports

As mentioned earlier, the reports included in the reporting supplement and described in this Tech Tip were designed as illustrative examples of how you can easily create a weighted metrics system for your organization using Visual Intercept. Based on your own best practices you may want to create your own f(i) and f(p) that are different from what is included in these reports. Since each of these reports was created using Crystal Reports, it's easy to open any of them in Crystal and change the formulae. The file names for the Report for Incident Submission Metrics, Report for Incident Prioritization Metrics and the Report for Incident Throughput Metrics are, respectively, viIncidentMetricSubmission.rpt, viIncidentMetricPriority.rpt, and viIncidentMetricThroughput.rpt.

You can also add your own reports to Visual Intercept, allowing you to choose your own metric reports from the report format selection in Visual Intercept. If you would like more information on adding your own reports to Visual Intercept, please contact Elsinore technical support services at: support@elsitech.com.

Other Uses for Weighted Metrics

In addition to using weighted values in reports, the weight values are also exposed in the Visual Intercept object model. This means that you can write VBA macros or use the Visual Intercept Software Developer's Kit (SDK) to run tasks based on incident weight values. An example of this type of project would be an early warning service written using the SDK that would run your weight calculations per project and notify project leads if the f(p) reached a certain threshold.

Appendix: Sample Weight Values

These are the incident status and weight values that are included in the Visual Intercept example data and the script used to create new Visual Intercept databases. These values were set to provide examples of the type of weight values that would be useful within the context of the metric reports distributed with the product.

Status

Note: In these particular reports the status weight values were interpreted as 1 - (the weight value). See the section "The Incident Status as a Special Case" for a full explanation.

Name	Weight
Closed	1.00
Development	0.30
Duplicate	1.00
Investigation	0.20
N/A	1.00
New	0.00
Not Reproducible	1.00
Open	0.10
Q/A	0.60
Rejected	1.00
ReOpen	0.10
Suspended	1.00
System Test	0.50
Unit Test	0.40

Priority

Name	Weight
High	0.75
Low	0.25
Medium	0.50
N/A	0.00
Urgent	1.00

Severity

Name	Weight
Annoying	0.25
Confusion	0.50
Crash	1.00
N/A	0.00
Unexpected	1.00

Category

Name	Weight
Change Request	0.50
Configuration	0.50
Cosmetic	0.25
Defect	1.00
Design Flaw	1.00
Documentation	0.25
Education	0.10

Feature Request	0.50
Integration	0.50
Internal	0.50
N/A	0.00
Requirement	1.00
Unknown	1.00