With company valuations in the dumpster and credit markets tight, the economic crisis has forced hard looks at revenue sources and spending across the board. The economy aside, old ways of doing business are continuously challenged by competitors with innovative business models enabled by the internet.

These and other socio-economic trends are pressuring finance teams everywhere to make planning systems more immediate and responsive. The old way of thinking—you run a company through the once a year budget process—is on the way out in favor of new concepts and system approaches generally described as "continuous planning". The practical implementation of continuous planning is through rolling forecasts—frequent updates of financial plans that include integrating actuals data and rolling the forecast time period beyond the current fiscal year.

This white paper explores the concepts and practices of rolling forecasts with a focus on defining application requirements for integrating actuals, financial modeling and scenario analysis—the three foundations of a rolling forecast system.

Alight Planning is an integrated planning and analysis software package that supports the requirements for budgeting, forecasting and financial reporting.

Application Requirements for Rolling Forecasts

By Rand Heer and Ben Lamorte

Beth Anders, CFO for Whitehorse Corp., a $300 million manufacturer of electronic equipment, shook her head. The Ops Review meeting, all two hours of it, was a bomb. Peter Forrester, the Planning Director, was looking glum.

Despite intensive preparation, Beth and Peter's presentation of the rolling forecast left too many unanswered questions about where and how Whitehorse should cut back operations in light of a recent drop in orders and rising inventories. Like most manufacturers in their industry, the credit crunch had an immediate ripple effect that required action.

Preparing for the meeting, Beth and Peter realized that the company's 2010 budget approved only three months earlier was useless. Whitehorse had missed Q1 sales and profit targets by a wide margin. No amount of number crunching produced a credible forecast where Q1 misses could be made up in later quarters. This was a first at Whitehorse, coming off the full year budget at the end of Q1.

The real pain, however, came from working with the company's new budgeting software. Installed the previous summer, Whitehorse replaced its Excel budget templates with a high end budget application built on an OLAP (on line analytical processing) database. The new system had seemed to work: cycle times for budget revisions were reduced; line managers were less confused over how to input data; and consolidations were faster with fewer errors.

Beth and Peter had expected that the benefits would continue as they used the budgeting package for developing a rolling forecast at the end of Q1. However, the new system failed to support several critical needs. Here were the issues:

- **Integrating actuals.** Peter was able to import actuals for January through March into the budget package and report variances against budget. However, mechanics were not available for integrating actuals data into the balance of year plan. For example, he could not set up a model where Q1 actual orders drove Q2 revenues, nor could he spread Q1 actual spending rates into Q2 forecast months.
- **Modeling inventories.** The budget package included modeling tools. For example, during budgeting Peter could forecast inventories in the aggregate based on links to cost of sales and assumptions about inventory turns. However, for the new forecast he needed more granularity, a model that projected inventories by product group and type—raw, WIP and finished goods. The modeling in the budgeting application had an Excel-like formula interface, but it was too restrictive and difficult to use. Forecasting inventories at a more detailed level didn't get incorporated into the forecast. That was an embarrassment at the Ops Review.

Beth and Peter needed response times measured in seconds, not minutes or hours…They needed a tool that made collaboration a real time process.

- **Setting up scenarios**. While budgeting, Peter used the new system's versioning capabilities for managing revisions. It worked. For the rolling forecast, however, it didn't. The focus this time was on scenarios, also known as "what if" analysis*. The problems were maintenance. Peter couldn't make structure changes, like adding a line item, across multiple scenarios at one time. As well, if he needed to make a change to an input assumption across scenarios—like updat-

* We use the terms scenario and "what if" interchangeably. Both have essentially the same meaning—namely, an alternate financial analysis with its own assumptions that are different from a known starting point or baseline case.

ing rent expense—he had to change each scenario one by one. When comparing scenarios, Peter could compute variances only at the account level. There was no visibility into differences between scenarios of underlying activity drivers. In short, the new budget package didn't work well for the in-depth scenario analysis that Whitehorse needed.

- **Update response time.** The budget application's client server architecture and user security made gathering user inputs easier than with spreadsheets. However, Whitehorse's current planning situation was not about re-doing the budget. The company needed to make short term decisions about staffing, inventory management and other elements of the business, each with major impacts on profitability and cash flow. During the plan analysis, the cycle time between changing input assumptions and then viewing the impact on financials took too much time. Given the sheer volume of analysis required, Beth and Peter needed response times measured in seconds, not minutes or hours.

The Need for Rolling Forecasts

With company valuations in the dumpster and credit markets tight, the economic crisis has forced hard looks at revenue sources and spending across the board. The economy aside, old ways of doing business are continuously challenged by competitors with innovative business models enabled by the internet. With instant access to product information and comparative pricing, customers rule as never before.

These and other socio-economic trends are pressuring finance teams everywhere to make planning systems more immediate and responsive. The old way of thinking—you run a company through the budget process—is on the way out in favor of alternate approaches such as *rolling forecasts*, a discipline for developing plans and decision making promoted by such organizations as the Beyond Budgeting Round Table and planning gurus like Rob Kugel at Ventana Research.

Pulling from the literature, the table below presents the differences between budgeting, the historical discipline, and rolling forecasts, an supplemental method for planning promoted in this white paper.

Budgeting Versus Rolling Forecasts

	Budgeting	Rolling Forecasts
Timing		
Frequency	Once a year	Often—event driven
Cycle time	Months/weeks	Days/hours/real time
Time horizon	Fiscal year	Rolling
Process		
Versioning	One size fits all	Multiple scenarios
Collaboration	Submission/approval	Real time consensus
Deliverables	Reports in binders	Decisions; action
Data		
Type	Financial	Financial & operational
Inputs	Many direct	Activity/driver based
Measurement	Budget variances	Relative change
Level of detail	Precision driven	Relevant; material

Historically, budgeting has been the command and control foundation for running a business—that once a year Uber-process that produces a *one size fits all* binder document with performance targets, bonus formulas, sales quotas, headcount and spending controls, standard costs for inventory valuation, capital budgets, and Wall Street guidance—all based on the same numbers with the same underlying assumptions. The budget also becomes the company official forecast for at least two quarters because no manager admits to not being able to "make budget" sooner than Q3. That's the politics of budgeting.

Few finance organizations are ready to dump budgets *per se*. However, many are

looking critically at their spreadsheets or canned budgeting applications asking how they can deliver more immediate and responsive financial plans. In particular, finance is getting more serious about rolling forecasts that incorporate a senior level involvement in core operational issues and their financial impacts.

Many organizations already do rolling forecasts, principally Fortune 500 companies where the discipline is ingrained. Here, the challenge for finance is to find ways to make the process less political, especially in dealing with hockey stick forecasts.

Requirements for Rolling Forecasts

As Beth and Peter discovered, there can be a big difference between a budgeting application and what's needed for real planning—in their case, implementing a rolling forecast process that's heavily focused on modeled elements, scenario analysis and near term decision making.

Whitehorse's dilemma sets the stage for the theme of this white paper: what are the most important functional requirements to look for in a software package for implementing rolling forecasts.

As Beth and Peter discovered, there can be a big difference between a budgeting application and what's needed for real planning, in their case a rolling forecast…

In the materials that follow, we walk through three software feature/function areas that are central to all planning applications, whether simple budgeting or more sophisticated planning. Not so coincidentally, these are areas where Beth and Peter had problems preparing for the Ops Review meeting that bombed.

- **Actuals**—the issue is not whether an application can import actuals data; all of them do. The issue is how well imported actuals can be *integrated* into the forecast plan.
- **Modeling**—the issue is not whether an application includes modeling tools; all of them do. The issues are how flexible are the tools for building complex driver-based models and how easy is it for the super user, people like Peter, to build *and* maintain the models.
- **Scenarios**—the issue is not whether an application supports versions and scenarios; again, they all do. The issue is maintaining the scenarios, visibility into the underlying activity drivers of each scenario, and response times.

The materials that follow describe these issues and show how Alight Planning handles the functional requirements.

Integrating Actuals

Virtually all budgeting applications support importing actuals from the general ledger, typically based on rigid chart of accounts structures. As well, you can usually import headcount and salaries from the HR system. Once imported, budget applications also support comparing actuals to budget based on levels in the chart of accounts with computation of amount and percentage variances.

Outside the safety of the accounting structures, however, lining up actual and plan data in a continuous format for a rolling forecast is typically a zoo, principally because it is difficult to get actuals data apples-to-apples with plan data at the line item level.

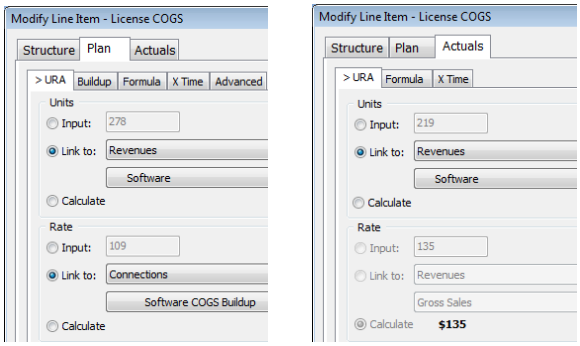
For example, for basic revenue planning:

- ✓ Units * Sell Price = **Sales Amount**
  500 * $100 = $5,000

For actuals, however, you get the data wherever you can, and the data is typically amount and units. Therefore, for an apples-to-apples comparison across units, price and amount, the actuals data must be modeled to back calculate the sell price:

- ✓ Sales Amount / Units = **Sell Price**
  $5,000 / 500 = $100

As the examples illustrate, actuals data frequently need modeling different from plan data to get valid comparisons and trends. For maximum flexibility in setting up rolling forecasts, the planning application should allow modeling of actuals with algorithms and linking separate from modeling of plan data.



Alight includes separate tabs for modeling actuals and plan at the line item level. In the example, plan units * rate = amount. For actuals, amount / units = rate.

In addition, actuals data need to be imported *from any source at any level of detail*, not just the general ledger. In the above revenue example, the actual sales amount is imported from the general ledger, but units for back calculating average price are typically imported from a data warehouse or a CRM system.

Finally, in our Whitehorse story Peter discovered with the budgeting application that he *could not* set up a model where Q1 actual orders drove Q2 planned revenues, a severe limitation. For robust rolling forecasts, you must have the capability of truly integrating actuals into the financial plan. That means: 1) being able to model actuals as an activity driver for the rolling forecast, and 2) spreading actuals data into plan time periods—e.g. projecting last month's actual spending run rates into future months.

		Nov 08 Act Units	Dec 08 Act Units	Jan 09 Units	Feb 09 Units
Products					
Product A	URA	950	780	900	1,000
Product B	XT	925	887	950	780
Total Products		**1,875**	**1,667**	**1,850**	**1,780**

In Alight, you can create data relationships where actuals are activity drivers for plan. In the example, Product A unit sales drive Product B sales with a two month time lag. Note that Product B's forecast sales for Jan 09 are the same as Product A's actuals for Nov 08—i.e. actuals are integrated into plan.

	Feb 2009 Act Amount	Mar 2009 Amount	Apr 2009 Amount
Services [6140]			
Legal Fees	$ 1,250	$ 1,250	$ 1,250
Accounting Fees	$ 950	$ 950	$ 950
IT Service Fees	$ 1,100	$ 1,100	$ 1,100
Payroll Service	$ 456	$ 920	$ 930
Total Services [6140]	$ 3,756	$ 4,220	$ 4,230

In Alight, you can "spread" actuals data into plan timeframes. In the example, Feb 2009 actual spending rates for Legal, Accounting and IT Service fees are spread into the plan time periods Mar and Apr 2009. The spread action on multiple line items is done in a single operation.

Driver-Based Modeling

Budgeting focuses on gathering static inputs from users, principally for headcount and expenses. Little modeling is involved—typically just calculation of payroll taxes and benefits.

Such a process is not workable for rolling forecasts. The cycle time for completing a forecast after month end close is too tight to accommodate broad based user involvement on the scale done for budgeting. As well, user based planning involves high vo-

lumes of static inputs in order to achieve precision. By contrast, rolling forecasts need to reduce the volumes of data, focusing instead on fewer truly material forecast elements.

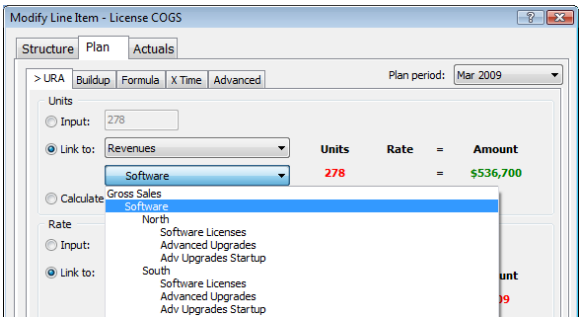
The objective of reducing data volume is achieved in part through "driver-based planning", a type of financial modeling where the most material items in a financial plan are linked to operational drivers or quantifiable activities of the business —e.g. volume measures such as units, transactions, subscribers, customers, call levels, hours, installations and the like.

Such driver-based modeling has three focuses: 1) revenue forecasting based on the volume measures and driver relationships *between* products, 2) variable and semi-variable headcount and expenses driven by the volume measures or identified underlying activity levels; and 3) for cash planning, balance sheet items such as inventories, accounts receivable and accounts payable, each based on their respective P&L drivers at a relevant level of detail.

Companies moving to a rolling forecast discipline should closely evaluate the modeling environment of the planning application they are buying into. Modeling and model maintenance should be fast, flexible, and (unlike Excel) provide clear visibility into linking relationships.

In our experience, there are two make or break functionalities in the modeling interface needed to make driver-based planning work:

- **Visibility into driver relationships.** This requires *object based linking* where the modeler establishes data relationships based on the names of things—e.g. *Commissions* are linked to *Net Sales* and the *Commission Rate* is 5%—versus cell based linking in Excel with formulas like = *Sales! C45 * L15*. Object based linking makes auditing and visibility into activity driver relationships significantly easier to track, thus eliminating many errors, speeding up development and reducing maintenance time.



Alight's modeling interface is based on object-based linking where you create data relationships by linking to the names of other line items or totals. Linking automatically operates across all time periods—i.e. no fill right as required with spreadsheets.

- **Modeling across dimensions.** Planning applications should support custom dimensions—e.g. product, customer, channel, region, project, job type, grade level, etc. The modeling environment should then allow tapping into the custom dimensions for building specialized activity-driver models—for example, aggregating sales line items by channel to drive channel specific discounts; aggregating employee salaries by job class for driving tiered bonuses. Applications without modeling based on dimensions do not have the flexibility required for driver-based planning.

Robust Scenario Analysis

Budgeting is about managing versions, not scenarios. As the budget is developed, finance keeps track of versions to understand who changed what and to make sure the right amounts are approved. Once a version is superseded, there is rarely a need

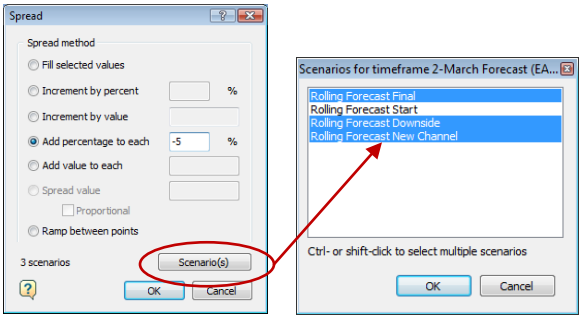
to look back at the old numbers or change them.

By contrast, rolling forecasts are about scenarios, lots of them. If you can't predict the future, the next best thing is to set up scenarios that let you explore how you might behave (or decide) if things are better or worse or just different. Unlike budgeting where you care about who changed what number, scenario analysis is about understanding what's behind the numbers—the most critical assumptions, volume and rate impacts, and especially what's driving material changes to the P&L and cash flow.

The deliverable of scenario analysis is *actionable knowledge*. By analyzing a specific scenario and comparing it to a baseline case or other scenarios, the management team is better able to evaluate best courses of action. Where there is an immediacy to the issues—e.g. to proceed with a capital project or change pricing—the deliverable is decision making. Because it's decision and action focused, robust scenario analysis is the most critical underpinning of continuous planning.

The functionality you need for effective scenario analysis goes beyond simple budget versioning. Here are our criteria:

- **Real time feedback**. Whether you're a financial analyst working through the numbers late at night or the CFO answering questions live in an operations review, scenario analysis should be delivered by the planning tool in real time. That is, when you change a value, all elements of the financial model—the P&L, balance sheet, cash flow, financial ratios, performance metrics—should update in seconds, not minutes or hours. In short, scenario analysis must satisfy *the need for speed* we're used to with Excel. Scenario analysis must be an interactive process responsive to questions and testing of assumptions on-the-fly. Continuous planning needs tight feedback loops on the numbers.



Unlike budgeting apps, in Alight you can change values of line items across multiple scenarios. Clicking the Scenario button lets you choose which Scenarios you want to apply the new values to. Updates to financials are immediate.

- **Maintenance *across* scenarios**. Budget versions don't require ongoing maintenance because old versions are superseded by new ones. Scenarios do. The planning application should support adding, modifying and deleting line items across selected scenarios in a single operation. Calculation and update of financials after structure changes should take only a minute or two, at most.

- **Robust comparison at the line item level.** Budgeting focuses on amounts in accounts. Rolling forecasts are about in depth comparison of scenarios and differences in values at any level of detail, especially at the line item level where the most significant inputs and modeling occur. Where the underlying data or links are available, scenario comparisons should reveal variances in underlying unit activity drivers and rates.

COMPARISON OF SCENARIOS WITH VOLUME/RATE CAUSAL ANALYSIS: Alight lets you compare scenarios at any level of detail including analysis of underlying units, rates and amounts. In the report below, the scenario called 2009 Downside is compared to the scenario 2009 Base Forecast. Each scenario has underlying monthly data. The right hand column is a causal analysis that computes by line item the amount variance between scenarios and the volume and rate components of the amount variance.

	2009 Downside			2009 Base Fcst			Volume Rate		
	Units	Rate	Amount	Units	Rate	Amount	Units	Rate	Amount
Gross Sales									
Software									
North									
Software Licenses	858	$2,029	$ 1,741,296	889	$2,212	$ 1,966,356	$ (62,914)	$ (162,146)	$ (225,060)
Advanced Upgrades	498	$1,373	$ 683,930	527	$1,501	$ 790,880	$ (39,827)	$ (67,123)	$ (106,950)
Total North	1,356		$ 2,425,226	1,416		$ 2,757,236	$ (102,741)	$ (229,269)	$ (332,010)
South									
Software Licenses	586	$2,069	$ 1,212,393	628	$2,250	$ 1,413,033	$ (86,895)	$ (113,745)	$ (200,640)
Advanced Upgrades	302	$1,395	$ 421,230	318	$1,519	$ 483,180	$ (22,317)	$ (39,633)	$ (61,950)
Total South	888		$ 1,633,623	946		$ 1,896,213	$ (109,212)	$ (153,378)	$ (262,590)
West									
Software Licenses	669	$2,027	$ 1,355,910	735	$2,213	$ 1,626,510	$ (133,767)	$ (136,833)	$ (270,600)
Advanced Upgrades	305	$1,373	$ 418,680	337	$1,498	$ 504,780	$ (43,927)	$ (42,173)	$ (86,100)
Total West	974		$ 1,774,590	1,072		$ 2,131,290	$ (177,694)	$ (179,006)	$ (356,700)
Total Software	**3,218**		**$ 5,833,439**	**3,434**		**$ 6,784,739**	**$(389,647)**	**$(561,653)**	**$ (951,300)**

- **Analytic tools**. Scenario analysis should incorporate related analytic tools such as:

  - ✓ **Sensitivity analysis:** Tools for identifying the most and least sensitive input assumptions and calculating their impacts on sales, profitability or any target in the plan.

  - ✓ **Goal seek:** Here's how: 1) choose a target such as operating profit 2) specify a financial goal—e.g. $1 million 3) select an input variable—e.g. unit sales. The application calculates how many units are required to achieve the $1 million profit goal.

  - ✓ **Causal analysis:** Where underlying units and rates are available, compute volume and rate variance impacts between scenarios. (See example above.)

  - ✓ **Dashboards:** Display key performance indicators that update as scenarios are changed. Allow display and rapid update of input assumptions from the dashboard.

Summary

Rolling forecasts require special features that extend beyond the capabilities of spreadsheets and many budget applications. Actuals data must be available for driving forecast values. Modeling tools must be robust and provide transparency into underlying activity drivers. Scenario analysis should be real time with maintenance across scenarios. It should include supporting analytic tools.

In a continuous planning environment, the return on investment is not from reducing time spent on planning. The ROI comes from better financial performance as a result of a more responsive planning environment and improved decision making.

Rand Heer is President of Alight LLC and the creative force behind Alight Planning. He was also the founder of Pillar and designer of Hyperion Pillar, the first enterprise software for budgets and forecasting.

Ben Lamorte is Vice President, Business Development at Alight. Previously, he headed his own consulting firm specializing in complex model building for such companies as Adobe, Kaiser and PlanetRX.

Web: AlightPlanning.com/Tel: (530) 622-5485.