Tap into the true value of analytics
Organize, analyze, and apply data to compete decisively
Preface

From the Editors’ Desk

**Analytics for a New Decade**
1. Post-Crisis Analytics: Six Imperatives 05
2. Structuring the Unstructured Data: The Convergence of Structured and Unstructured Analytics 13

**Revitalize Risk Management**
3. Fusing Economic Forecasts with Credit Risk Analysis 21
4. Unstructured Data Analytics for Enterprise Resilience 29
5. Why Real-Time Risk Decisions Require Transaction Analytics 37

**Optimize to Drive Profits**
6. Ten Questions to Ask of Your Optimization Solution 47
7. Practical Challenges of Portfolio Optimization 55

**Understand Your Customer**
8. Analytics in Cross Selling – A Retail Banking Perspective 61
9. Analytics as a Solution for Attrition 69
10. Customer Spend Analysis: Unlocking the True Value of a Transaction 77
11. A Dynamic 360º Dashboard: A Solution for Comprehensive Customer Understanding 85

**Fight Fraud More Effectively**
12. Developing a Smarter Solution for Card Fraud Protection 93
13. Using Adaptive Analytics to Combat New Fraud Schemes 103
14. To Fight Fraud, Connecting Decisions is a Must 109

**Improve Model Performance**
15. Productizing Analytic Innovation: The Quest for Quality, Standardization and Technology Governance 117

**Leverage Analytics Across Lines of Business**
16. Analytics in Retail Banking: Why and How? 125
17. Business Analytics in the Wealth Management Space 135
Productizing Analytic Innovation: The Quest for Quality, Standardization and Technology Governance

The use of predictive analytics is becoming ubiquitous within modern financial IT solutions. However, with the increased complexity of analytic offerings comes the need for a standardized process and build methodology across all analytic model development. The software engineering profession, facing nearly identical constraints, has created governance methodologies that are re-usable for analytic development. As such, the quest of the analytics manager is two-fold:

1. Apply these effective methodologies to standardize analytic development. This will yield dramatic improvements in the quality and consistency of an analytic team’s output, while enabling the flexibility needed for analytic innovation.

2. Have analytic technology governance in place to ensure that analytic innovations comply with analytic development standards and constraints.

Introduction

As predictive analytics becomes a ubiquitous critical component of modern financial IT solutions rather than a state-of-the-art add-on feature, analytic development managers are faced with new challenges—ones that go beyond concerns with data quality, choice of training techniques, latest analytic research, and model performance.

Mission-critical business applications operate under many constraints, including stability in time, robustness under unexpected conditions, and scalability to handle the actual business information flow. These constraints apply equally to application software as well as to analytics within the software. Analytic model development across all applications needs to follow a similarly well-defined development methodology and process.

Customers of analytics, particularly top financial institutions, are increasingly aware of these complexities and possible implications. These customers require more transparency that an analytic team is utilizing a standard/governance with regard to model-building methodology, technology and process.

This article describes potential solutions and focuses on the practical aspects necessary to transform an analytic product organization’s methodology to enable successful delivery, deployment, and operation of hundreds of
predictive models annually. The model development standards and governance outlined in this article are essential components within a larger set of analytic requirements for the business, which may include legal and regulatory requirements.

Software engineering has matured over time by developing processes and methodologies that help ensure consistency and quality of a software release. These methodologies, which ensure that software projects are delivered on time and within budget, started being developed as early as the 1960’s. Since then, they have continuously been improved to meet the demands of the growing size and complexity of the software products. These methodologies have played a pivotal role in the success of today’s software industry.

The software development methodology has changed from structured programming to object-oriented design and development to service-oriented architectures and cloud computing. The process approaches evolved from traditional waterfall to a variety of iterative approaches—such as Rational Unified Process, Extreme Programming, and Agile Unified Process—developed in recent decades.

Analytic development can benefit from adapting many aspects of software development, such as:

- Use of version control and coding standards
- Configuration and release management
- Development of test cases and automated test suites
- Code modularization and sharing (for example, variable libraries)
- Early prototyping and integration (for example, development of skeleton models for deployment validation)

- Iterative approach to the model development cycle

Tools Governance

Development of robust and high quality analytics includes the need for strict adherence to a standardized, approved, and tested tool set. Well-tested tools are paramount to the correct development of the model and confident use of the analytic model in business applications.

For example, a sampling script that inadvertently changes the training data set by excluding an important component of the population makes the final deployed model not representative of the production environment.

While one can rely on established third-party packages—such as R, SAS, SPSS, or FICO™ Model Builder—for most typical statistical calculations, in practice, there is a large amount of custom code written by the analytics development team. This is needed to implement specific custom adaptations of cutting-edge algorithms, perform ad-hoc data transformations to prepare the data for model development, implement specific intellectual property algorithms, and combine individual calculations into automated modeling steps.

Without a centralized process, ad-hoc tool development results in accumulating too many tools developed by different scientists, having multiple versions of the same tool (sometimes with subtle but important functionality differences), inconsistent documentation and interfaces, and duplicate or redundant functionality. Further, some functionality affecting the resulting model quality and performance may be based on arbitrary individual choices and not fully reviewed, tested and validated.
The solution is to have a well-defined tools governance process that determines all aspects of developing, testing, modifying, documenting, and distributing the common analytics toolset. This ensures consistency across many scientists developing analytic solutions. The process often includes coding guidelines and procedures for using version control, nightly builds and configuration management. It also covers organizing and documenting various third-party components (compilers, utilities, statistics packages, SDK's), and unifying the versions and locations of the software across different analytic sites and servers.

While many tools still originate from an ad-hoc solution to a modeling problem, under the tools governance process, any discovered gap or deficiency in functionality is brought to the Technology Governance Board to consider enhancement of an existing tool or creation of new standard tools.

Model Standardization

Even with appropriate tools, it is not possible to ensure high model quality if models are being developed without some form of common standard and code sharing. Model standardization is necessary to facilitate the propagation of improved or corrected variable definitions and bug fixes, and to ensure compliance with the latest data specifications and consistency of the model external interface (for example, reason and error codes are consistent across models).

Model standardization also promotes collaboration between project teams who contribute to the standard. This reduces duplication of efforts and thereby leaves more time for analytic innovation projects.

The most important aspects of model standardization are model code sharing and creation of a shared variable library. A standard model code structure and organization should exist to facilitate sharing, while leaving the possibility of customizing variables for an individual model—for example, by introducing country and region specific functions and variable definitions.

Code sharing is essential for achieving consistency in how a model handles typical scenarios and use cases. Response generation functions, common data fixes, model upgrade logic, error handling and error codes can all be implemented in a shared codebase utilized by individual models. The shared code library is accompanied by its own automated test suite that invokes common test cases, making it possible to validate the shared code before it becomes part of any model build.

In analytics development, a shared variable library stands out from other shared code since it encapsulates the accumulated knowledge and Intellectual Property (IP) of the analytics team. Variables are the quintessence of the model. Hence, it is paramount for variable definitions to be shared in a version control repository, where variables are tested for correctness of the definition and leverage consistent variable definition standards. Variables are constantly reviewed for improvement and, through a common model standard, can be easily made available to all models.

Variable standardization and version control also allow for proper inventory of variables. This prevents redundancy of variable definitions for those that are closely related, and also allows for specific feature detector development where variable sets may be smaller. Variables are logically classified based on the types of behaviors to which they respond.

For example, given a well-organized variables
library, one can easily determine the variable sets available for cross-border fraud detection and, if necessary, focus specifically on new variable definitions/improvements. Any improvements can then be available to all future builds, not just to a specific modeling project where the variable innovation work may have occurred.

**Modeling Process Standardization**

For an analytics development organization that delivers hundreds of models every year, it is essential to follow a standard methodology and process. Each individual model build follows this process, which defines the essential milestones and subsequent checks and balances that need to be in place throughout the model building process.

The process typically starts with Requirements Definition. Every model may have its own requirements for the types of input and specific detection targets. Also included would be supported versions of the data specification, and any customer and environment-specific requirements and constraints. These can be captured in an analytics tracking document, which defines both the requirements and the deliverables for the model build.

The requirements and data availability determine the choice of the model architecture and modeling technologies. These choices can be refined as the process goes iteratively through the following typical steps:

- **Data Analysis** – This task includes ensuring the proper acquisition and data quality of contributed data for a model build. This should consist of a standard Data Quality Report (DQR) that summarizes the data issues, and contains the basic statistics for all the input fields and sets of automated alerts to data issues. These alerts need to be reviewed for go/no-go decisions on utilization of the modeling data in the build. The report is also used on a recurring basis as a tool to tactically improve data contributions where strict adherence with the data specification/standards is lacking.

- **Data Preparation** – Once high quality data is collected, the data continues to be analyzed from the model build perspective. At this stage, more subtle issues (such as intersection between various data streams and temporal shifts in distributions) are reviewed with the analytic manager. Additional sampling and pre-processing is usually performed based on quality and composition of data from different sources (for example, different institutions or sub-systems).

- **Variables Definition and Generation** – This involves writing code to calculate the variables, striving to provide computationally efficient and statistically robust (to errors, outliers, and temporal shifts) definitions. Attention must be paid to potential legal and regulatory constraints—such as the Fair Credit Reporting Act or non-discrimination laws that may limit or prohibit the use of certain input fields. Variables that are part of the model standard have fully automated test suites and are approved for use in models. New variables are reviewed by the Technology Governance Board for approval in a build, but also for future inclusion in the model standard codebase.

- **Variable Selection** – This is usually done using a variety of techniques—such as correlations, regressions, mutual information, and sensitivity analyses. Specific review thresholds related to sensitivity are used to point to any problematic variables that could indicate issues—such as target leaks or spurious correlations.
Model Training - This presents challenges, since many training technologies, such as neural networks, are inherently complex and involve difficulties in explaining the results, parameter optimization and tuning, possibility of overtraining, and instability to initial model weights values. Standard stopping criteria and multiple validation data sets are used and reviewed to ensure robustness.

Evaluation and Validation - These techniques are not straightforward since they usually involve multiple population segments and data from multiple sources. Final validation involves using a hold-out dataset that has not been used in any way during training, as well as techniques such as out-of-sample testing, cross and leave-one-out validation.

Model Creation - This involves creation of a deployable model, usually by means of automated tools that convert the raw output of a training algorithm into deployable model code. Many important steps (such as model response calibration and input capping, continuity and functional testing) are performed prior to delivery of the final model.

Analytic models should be treated like a software release, and subject to complete functional testing as it goes through the software integration and Quality Assurance (QA) process.

Since the final model is essentially executable code, bugs may go undetected on modeling data, but present themselves when the model encounters the real production data and environment. Similar to software testing, functional test cases are developed and incorporated into an automated test suite. These often involve using synthetic data (including erroneous data) designed to expose all potential execution paths and error-handling scenarios in the model code.
The tests include model upgrade and score continuity tests to ensure that the model upgrade will not adversely impact the operational environment.

To facilitate seamless model integration, a Model Delivery Standard is utilized. This standard defines and documents all the interfaces between the software and the model, as well as the accompanying test data and documentation.

Analytic Innovation and Technology Governance

To address a maturing analytic development methodology, an Analytics Innovation and Technology Governance Process must exist to ensure that only fully validated and approved technologies are used in mission-critical and productized analytics. This process takes innovations from concept to productization, and standardizes the technology prior to adoption in individual models.

The basic principles of a Unified Process (such as RUP or Agile UP) developed in software engineering can be successfully adopted in analytic development, especially with regard to managing innovation projects.

A Unified Process defines four project life-cycle phases. It emphasizes iterative development where different project activities—such as requirements definition, design, coding, and testing—are mixed in different proportion at each project phase.

- Inception Phase – This phase is based on an initial idea that often results from a new requirement, a discussion with the customer, an observation during model development, or a research article describing a new algorithm. The inception phase involves establishing an initial set of requirements for the innovation, initial design, and experimentation. This results in a high-level project definition, often formally documented in an Analytic Tracking Document where stakeholders from different departments agree on the project requirements, scope and budget.

- Elaboration Phase – In this phase, key assumptions are validated. The development work focuses on the areas of high uncertainty and risk through validating the design. This phase involves finalizing the design, obtaining key experimental results and code prototyping. Any significant change in design/approach is updated in the Analytic Tracking Document and requires re-approval before moving to the Construction Phase.

- Construction Phase – This phase focuses on developing the final version of the analytic technology based on the results from the Elaboration Phase. The production code is written, the necessary modeling steps are identified, and new test cases are developed for the technology. This analytic technology is then ready for final evaluation and incorporation into the model standard.

- Transition Phase – The delivery from the construction phase is reviewed by all stakeholders with respect to the value proposition of incorporating the analytic technology into the standard product suite. If the innovation meets approvals, a roadmap item is established to determine when work will be performed to incorporate the innovation into the model standard. Once in the model standard, the analytic innovation is available for use across all analytic development, the modeling process is modified accordingly, test cases are integrated into the test suite, and any necessary new tools are added to the Standard Toolset.
Analytic Development Methodologies are maturing. Mission-critical business applications call for high performance analytic models, but in equal measure, robust, stable, and predictable execution in their production environment. This requires that the Analytic Development Methodology adopt Tool Governance/Version Control, Model Standard, Model Process Standard, and Technology Governance for incorporation of formally approved, tested, and validated analytic innovations into the Model Standard.

To meet these requirements, leading analytics teams are exploiting Software Development Methodology practices and adapting them for Analytic Development. Clients are benefiting from these improved methodologies through consistency in the development process, consistency of model code, and stability of the models that drive their business decisions.

**Conclusion**

[Phases and activities in a typical innovation project](#)
FICO (NYSE:FICO) gives businesses an analytic advantage by using powerful predictions of consumer behavior to drive smarter decisions. Clients in 80 countries work with FICO to increase customer loyalty and profitability, cut fraud losses, manage credit risk, meet increasing regulatory and competitive demands, and rapidly build market share. FICO also helps millions of individuals manage their credit health through the www.myFICO.com website. FICO. Make every decision count.
### Global Presence

**Americas**
- Brazil: Navi Mumbai, Canada: Calgary, Toronto, Mexico: Monterrey
- United States: Atlanta, Bellevue, Bentonville, Bridgewater, Charlotte, Detroit, Fremont, Hartford, Houston, Lake Forest, Lisle, New York, Phoenix, Plano, Quincy, Westwood

**APAC and Australia**
- Australia: Brisbane, Docklands, Melbourne, Perth, Sydney
- China: Shanghai, Hangzhou
- Hong Kong: Central
- India: Bangalore, Bhuvaneshwar, Chandigarh, Chennai, Gurgaon, Hyderabad, Jaipur, Mangalore, Mumbai, Mysore, New Delhi, Pune, Thiruvananthapuram
- Japan: Tokyo, New Zealand: Wellington
- Philippines: Metro Manila
- Singapore: Singapore

**Europe**
- Belgium: Brussels
- Czech Republic: Brno, Prague
- Denmark: Copenhagen
- Finland: Helsinki
- France: Paris
- Germany: Frankfurt, Stuttgart, Walldorf
- Ireland: Dublin
- Italy: Milan
- Norway: Oslo
- Poland: Warsaw
- Spain: Madrid
- Sweden: Stockholm
- Switzerland: Geneva, Zurich
- The Netherlands: Amsterdam
- United Kingdom (UK): London

**Middle East and Africa**
- Kingdom of Saudi Arabia: Riyadh
- Mauritius: Rodrigues
- UAE: Dubai, Sharjah

For more information, contact askus@infosys.com

### About Infosys

Infosys Technologies Ltd. (NASDAQ: INFY) defines, designs and delivers IT-enabled business solutions that help Global 2000 companies win in a flat world. These solutions focus on providing strategic differentiation and operational superiority to clients. Infosys creates these solutions for its clients by leveraging its domain and business expertise along with a complete range of services.

With Infosys, clients are assured of a transparent business partner, world-class processes, speed of execution and the power to stretch their IT budget by leveraging the Global Delivery Model that Infosys pioneered.