Enterprise Financial Modeling

Actuaries' Club of the Southwest 2014 Fall Meeting

November 6, 2014

Jason Kehrberg, FSA, MAAA

PolySystems, Inc.
Actuarial Software & Data Solutions

Presentation Outline

- What is enterprise financial modeling?
  - Comparison to traditional desktop modeling
  - Evolution of actuarial modeling
  - Actuarial end-to-end processing
  - Steps to enterprise financial modeling
  - Model rationalization and fit for purpose
- Managing enterprise financial models
  - Modeling environments
  - Input and output management
  - Communicating model results
  - Model governance
  - Production and model change procedures
  - Modeling controls

What is Enterprise Financial Modeling?

What is Enterprise Financial Modeling?

- Financial modeling done in an centrally managed enterprise environment that integrates with the entire enterprise infrastructure and IT architecture standards
  - Better data management with quick, automatic access to quality data and consistent assumptions and modelling efficiency techniques
  - Enhanced security using permission-based development, test, staging and production environments
  - Production quality controls
  - Scalable distributed processing for improved performance
  - Improved automation capabilities
  - Integration with business intelligence services for reporting
  - Automatic backups and redundancy for robust 24/7 availability
  - Defined roles and responsibilities
  - Official change control and validation procedures

Desktop vs. Enterprise Financial Modeling

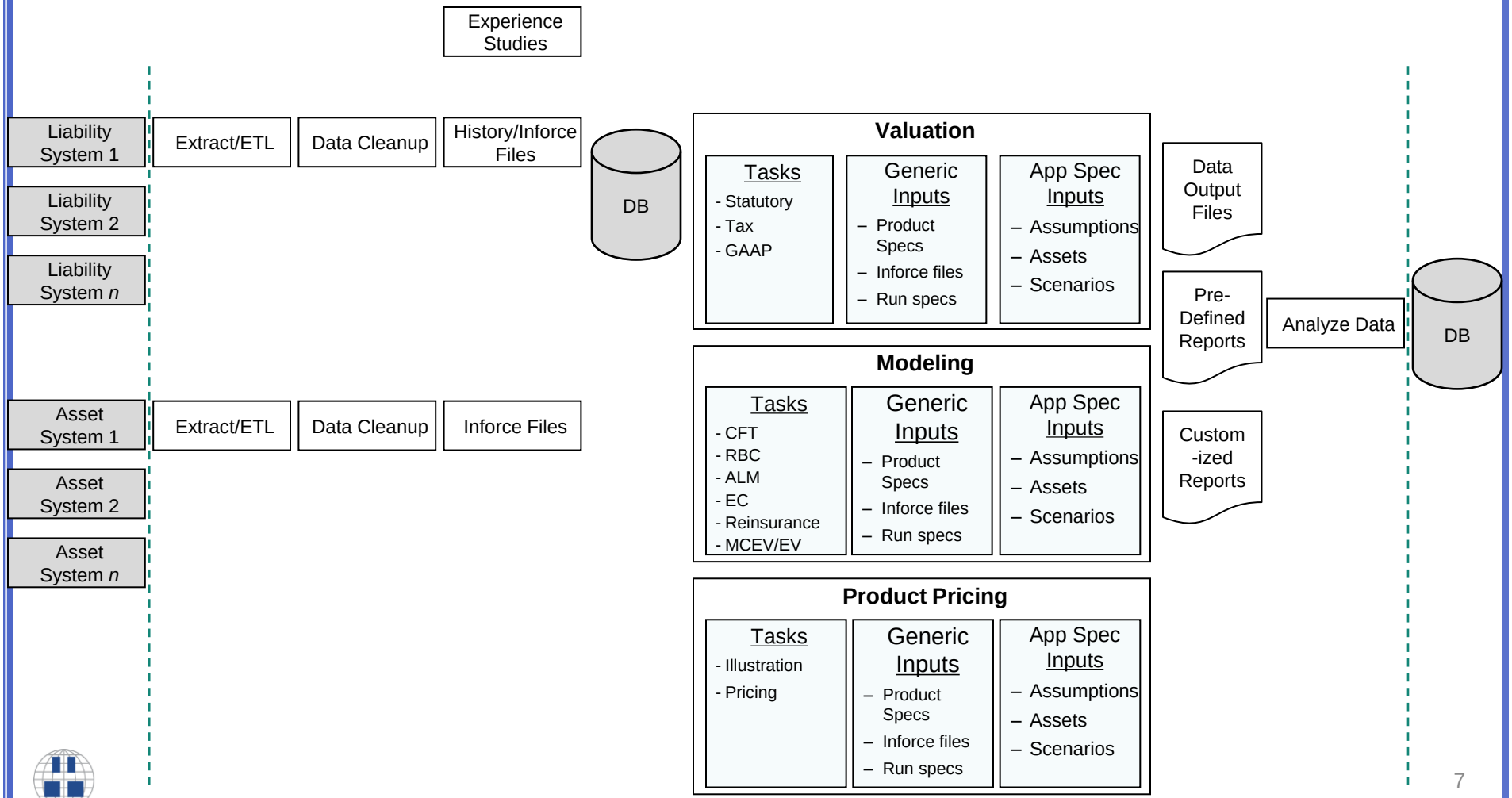
- Desktop
  - Focused on individual actuarial tasks/calculations
  - Ad-hoc, short term,
  - Siloed infrastructure with redundant data and models
  - Local security and individual user rights
  - Models for creativity and individual preference
  - Focused on a single model application
  - Actuarial independence from IT
  - Minimal governance and controls
- Enterprise
  - Broad processes
  - Governed and repeatable
  - Centralized infrastructure, shared data and reused models
  - Centralized security, roles, and audit trails for multiple users
  - Automated models for timely and efficient results
  - Results are consistent, repeatable, and reproducible, with shorter reporting timescales
  - Reduced risk
  - Grid scalable, reduced run time
  - Actuarial leverage of IT

Evolution of Actuarial Modeling

- Although pricing calculations, planning/forecasting, and formulaic valuations had been done for years, actuarial modeling of cash flows did not go main stream until asset adequacy analysis using cash flow testing was mandated by regulators in the 1980s
- Internal and external stakeholders have been adding to the modeling workload ever since: C-3 Phase I & II, AG43, PBR, various GAAP requirements, economic capital, embedded values, duration calculations, what-if analyses, etc.
- Most have remained as one-off desktop models, **even as complexity increases and results have a more direct and visible role in company financials**, never realizing the consistency and sustainability associated with enterprise processes
  - Easy to spawn and hard to kill (even during M&As), their number has multiplied over the years (much like spreadsheets)
  - Increased risk due to ever more frequently changing assumptions and inputs, while subject to increased internal and external scrutiny

Actuarial End-to-End Processing

- Actuarial department work spans from upstream admin system extracts to downstream final reporting



Steps to Enterprise Financial Modeling

- Executive sponsorship and a culture ready for modeling standards
- Model inventory and risk assessment
- Assess core actuarial cash flow projections across modeling platforms, rationalizing their use
- Assess roles, centralizing modeling work where possible
- Design and construct a modeling structure that works with enterprise infrastructure and IT architecture standards
- Design and set up the following *functionally separate* components on the new enterprise modeling structure, leveraging existing capabilities if possible, and satisfying each modeling application:
  - Data extracts, assumptions, product specs, and other input items
  - Core cash flow projections (use modular, object orientated programming)
  - Canned/custom reports and other output items
  - Controls, automation, and execution processes
- Perform baseline model testing/validation (initially and periodically)
- Review controls for production and model changes

Model Rationalization and Fit for Purpose

- Is the model good enough to do the job it was designed for?
  - "All models are wrong, some are useful"
  - "A model that is perfect in every respect is as useful as a map with a scale of one"
- Although model rationalization often results in fewer models and modeling platforms, the goal of "one corporate model" is often hard to achieve in practice
  - Models exist across multiple business units / geographies with varied products and jurisdictional regulatory requirements
  - Need for nimble, ad-hoc modeling in sandbox environment
  - "Closed" vs. "Open" systems
  - Vastly different requirements for time horizons, granularity, etc., can be hard to fit in one model
  - May want multiple models for validations, reconciliations, and/or redundancy
  - Multiple models increases the importance of good assumption governance for consistent assumptions

Managing Enterprise Financial Models

Modeling Environments

- Test environment
  - For controlled testing of approved model changes (i.e., new coding) and approved software updates
- Staging environment
  - Integrated with enterprise infrastructure and IT architecture for additional controlled testing on a production-like environment
- Production environment
  - After approval, models enter a “hands off” production environment
  - Locked down to prevent unauthorized changes
  - Integrated with enterprise infrastructure and IT architecture
  - Automated versioning for each change and reporting cycle
  - A post-production environment for ad-hoc analysis/adjustments?

Do your company have a production-grade modeling environment? For most or some modeling processes?

The Development Environment

- Starts with a download of the production model, controlled to mask any sensitive data
- For ad hoc analyses
  - Evaluating various reinsurance arrangements
  - Performing sensitivities, e.g. on newly proposed assumptions
  - Other “what if” analyses, e.g. senior management or regulator requests
  - Product development/pricing
  - Development of enhanced
- For controlled changes to the enterprise model
  - Model enhancements
  - Incorporation of new products and regulations

The processes for promotion to the next environment leverage enterprise security and version control tools.

Model Input Management

- As with any model, garbage in → garbage out
- Automated data extracts reduce the chance for human error
  - Inforce data from admin systems
  - Existing asset data from asset admin systems
  - Input from accounting or other valuation/modeling systems
  - Assumptions from centralized warehouse, including scenarios
  - Modeling efficiency techniques such as compression and representative scenarios can be automated within the ETL process or within the main calculation engine
- As with inforce files and assumptions, product specifications, run time parameters, and the underlying code itself are inputs to the process and should be locked down in a production environment
- Version control tools exist and can help with managing and archiving different iterations of the model and its input data

Model Output Management

- Model output should be linked to the input and model version used to create the output for control, reproducibility, and facilitating understanding of results
- Choice of output to discard, archive, and/or upload to data warehouse can vary by model application (e.g., baseline vs. sensitivity, deterministic vs. stochastic, frequency)
  - Maintain historical data for EGPs, experience reporting/studies, etc.?
- Back-end calculations (e.g., discounting, aggregating, CTE calculations, topside adjustments) should be done within the enterprise production environment
- Reporting tools
  - External vs. internal requirements
  - Audit and control reports, pre-generated canned reports, custom ad-hoc reports, attribution analyses, drill-down capabilities
  - Modeling software, Excel/Access, enterprise business intelligence services

Communicating Model Results

- Communication of model results must be reliably timely and effective in order for senior management to confidently report on them and use them for decision making
- Must be able to explain financial results at various levels of granularity, and why they have changed over time
  - Biggest challenge is understanding the drivers of movement
  - Step-by-step attribution analyses from one period to the next can help but is can be complex itself and may require multiple runs
- Tools for getting comfortable with and communicating results:
  - Trend, sensitivity, outlier, and other analyses
  - On demand “what if” analysis and drill down capabilities
  - Good controls and model governance
  - Assumption setting based on robust experience studies and good assumption governance
  - Smoothing mechanisms?

Model Governance

- Like all governance, it is an measured approach for controlled decision making and organized activity, primarily aimed at reducing risk
- A model governance document establishes the rules, roles, and controls related to models
  - Goal is to reduce model risk
  - A framework for model management
  - Applies to production, changes, and validation of accuracy
  - Includes architecture, coding and documentation standards
  - Model risk includes calculation errors, strategic errors due to bad models, bad assumptions/data, models not fit for purpose, and errors due to miscommunication of results
- It really comes down to execution and enforcement
  - Governance oversight committee to enforce compliance?
- Visibility of model governance is increasing under PBR

Roles and Responsibilities

- Best practice involves separation of duties (who does what where)
- Roles include: model steward, model architect, data steward, extract programmer, program developer, tester, actuarial “coder”, user of model output, power user, SME, software vendor
- Is there a dedicated model team?
  - Who makes the major modelling decisions and is responsible for signing off on model updates, production results, and model changes?
  - Does your company have a chief modeling officer? A model steward?
  - Who can request model runs and model changes?
  - Who runs the model and makes the changes? IT? Anyone?
  - What are the individual access rights?
- What responsibilities exist outside the model team?
  - What is the relationship with IT?
    - Use of service level agreements?
    - Integration with systems governance?
  - What is the relationship with the software vendor?
  - What is the level of integration with assumption governance?

Model Production Procedures

- Repeatable processes for updating and running the production model should be the same each cycle (absent any structural changes to the model)
  - Formats for input data
  - Default routines for moving to the next reporting cycle
    - Updating run-time parameters, inforce data, assumptions, etc.
  - Naming conventions
  - File structure
  - Modeling efficiency techniques such as compression and representative scenarios
  - Automation routines
  - Priorities for managing the production queue (including grid scheduling)
  - Validation procedures
  - Use of production controls

Types of Model Changes

- Various functional components of the model can change
  - The underlying code
  - The input (moving to the next reporting cycle, assumption updates)
  - The input specification structure (2D mortality tables)
  - Output items
- Causes of model changes
  - Significantly changes in product design or regulations
  - Changes to data sources
  - New dynamic assumption capabilities
  - Improved modeling of management actions and business logic

What is the level of model change governance at your company?

How frequently are changes made to your production models?

Model Change Procedures

- Normal change controls include access rights, automated logging of user/date/time/comments for each change, testing requirements, and the following controlled steps:
  - The change request, solution proposition and selection, programming and coding, software testing (unit, integration, regression), user acceptance testing (validation procedures and impact analysis), release and deployment
  - Multiple approvals/signoffs along the way
  - Priorities for managing the change queue
- Change controls can vary by model application
  - Pricing requires additional flexibility outside normal controls
    - Although ideally on the same platform used for financial results
      - May require some up front reconciliation
      - But helps avoid unexpected financial results down the road

Modeling Controls

- Controls are used to regulate model processes
- The use of controls in modeling continues to increase due to regulations and increased focus on model risk
- Controls will vary with the model environment, model application, and model software being used
  - Prevents unauthorized access to locked down production environment
- Controls will also vary depending on whether the process is a model update, model run, or model change

QUESTIONS