

# Bringing actuarial valuation systems out of Excel

May 27, 2015

# Objectives

- The nature of an actuarial valuation
- Excel
- WCB Alberta 10 years ago
- Determining the path
- System conversion
- Financial aspects
- Lessons learned
- Challenges
- The ultimate goal

# The nature of an actuarial valuation

- Complex data
- Layers of assumptions
- Actuarial methods
- Results are significant
- Actuarial SOP govern actuarial aspects
- Delivered through computer software

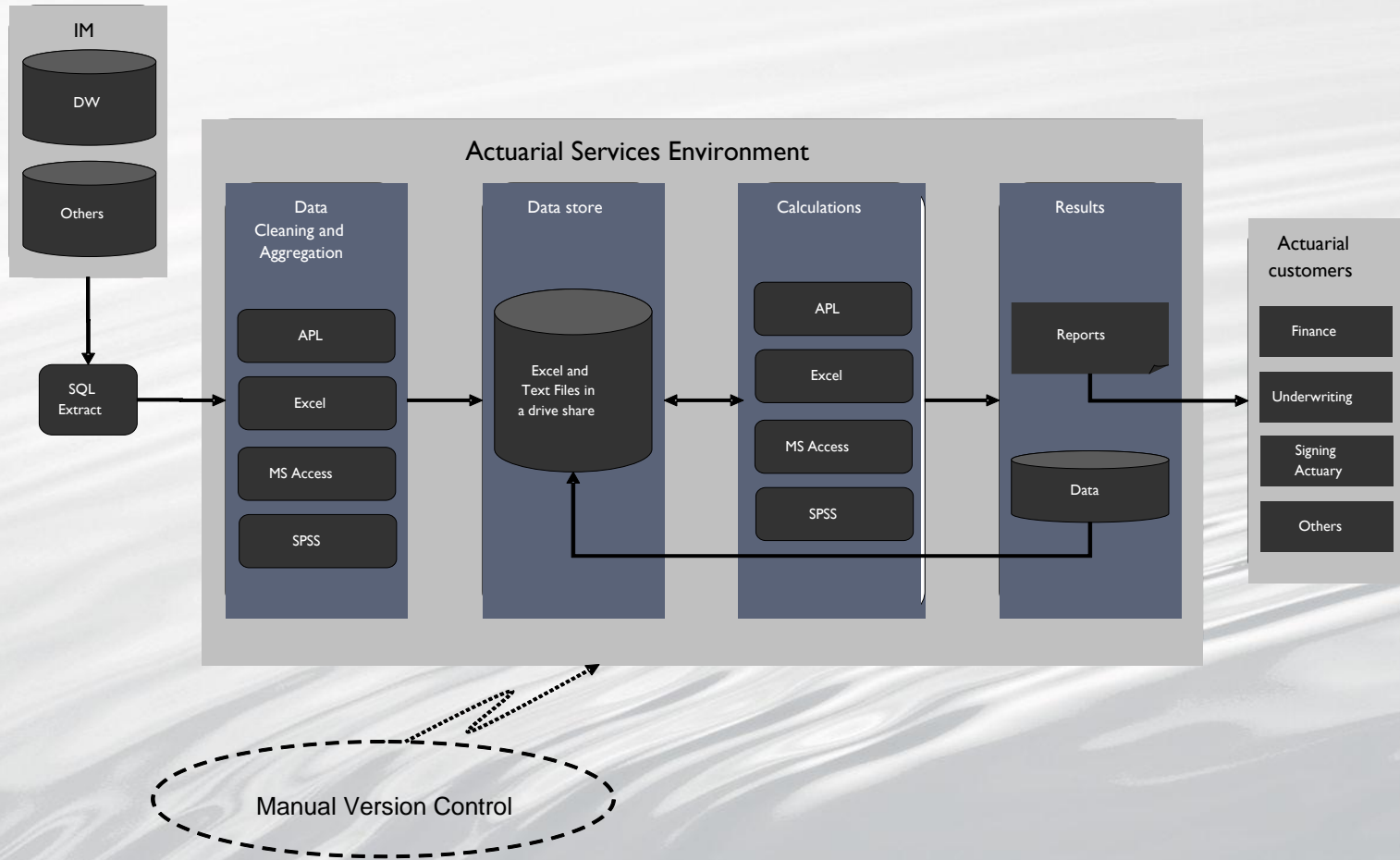
# Excel

- The most important software of all time
- Excellent for simple financial modeling
- Not sophisticated enough for an actuarial valuation
- Prone to error
  - Due to the nature of Excel
    - Data redundancy
    - Cross reference and links
    - Thousands of cells with embedded formulas
  - Human errors

# WCB Alberta 10 years ago

- A valuation was done using
  - 800+ Excel sheets
  - 100+ APL modules
  - and other programs
- Parallel work to minimize errors
- Key person situation
- “Too loose”
- Management letters from Eckler

# WCB Alberta 10 years ago



# Determining the path

- A system that can be used for
  - Valuation
  - Rate making
  - Other major actuarial work
- An IT consultant to look for an off-the-shelf product
  - Prophet
  - Moses
- Uncertainties
  - WCB environment
  - Customization and future maintenance

# Determining the path

## Eckler's review

- With both actuarial and IT background
- Understanding business objectives and concerns
- Reviewing the current process
- Identified alternatives to formulate a solution
- Identified two solutions
  - A commercial software (Prophet)
  - An in-house development approach (SQL Server)
- Need a full design process to implement an effective solution regardless of choice
- A proof of concept to study the feasibility



# Determining the path

## Proof of concept

- Implement a stripped-down valuation
  - one in-pay benefit
  - one outstanding benefit
- Results
  - Prophet
    - Two analysts, 2 months, no satisfactory result
    - Foresee significant amount of effort spent on customization
    - Not flexible in data manipulation
  - SQL Server
    - One analyst, five weeks, desirable results
    - Excellent at data manipulation, not the best choice for complex actuarial calculations
- An in-house approach would address most of Eckler's concerns

# System conversion

## Building blocks of an in-house solution

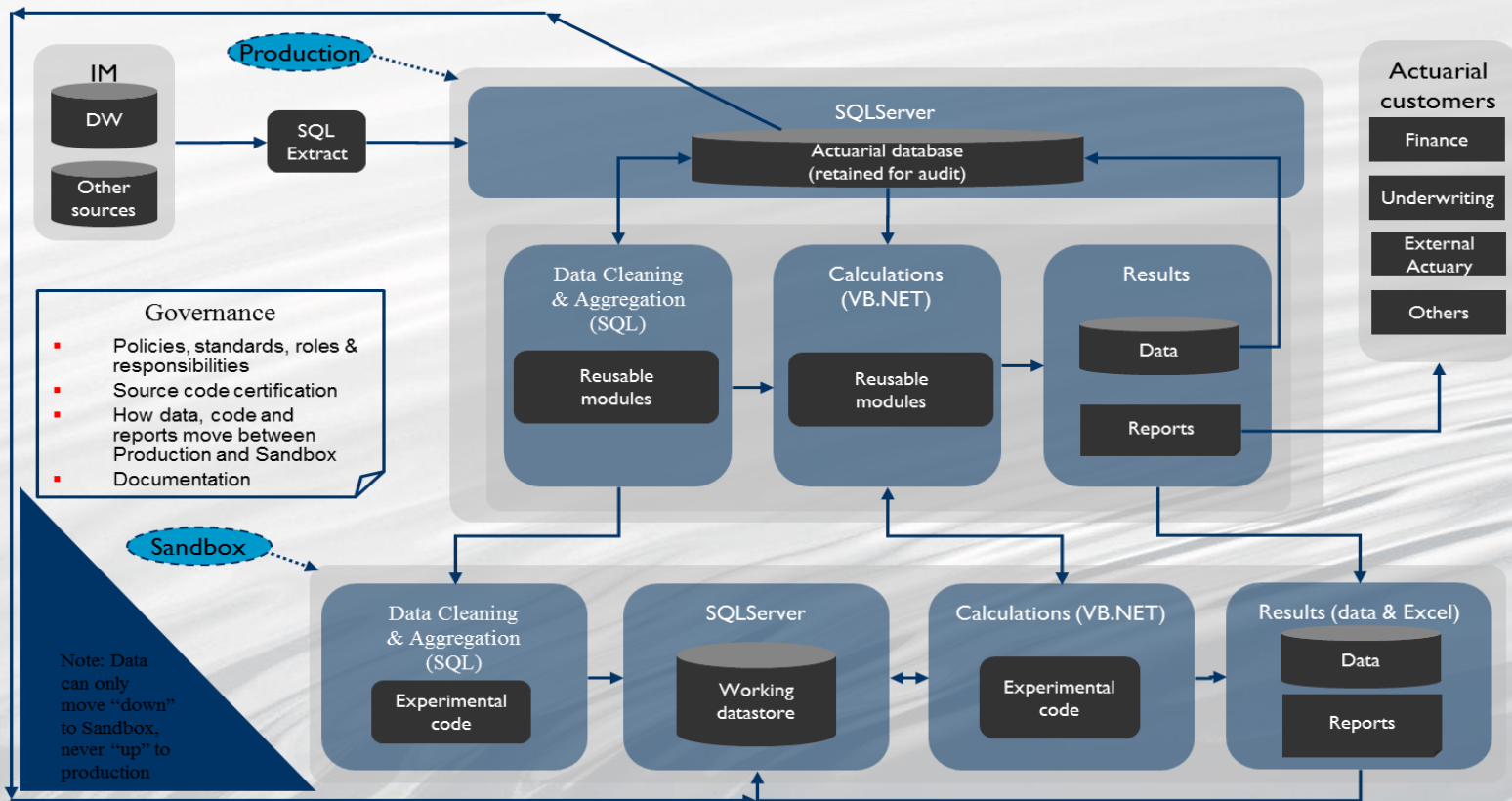
- Tools selection
- Governance model
- Conceptual design
- Implementation approach
- A roadmap

# System conversion

- Tools selection
  - MS SQL Server for the database
  - MS Visual Basic.NET for the development environment, and
  - MS Excel for reporting
  - MS Visual Source Safe for the version control
- Enhanced governance
  - Roles and responsibilities
  - Service level agreement with information management
  - SQL database standards
  - Coding best practices/standards
  - Documentation standards
  - Promotion/demotion standards
  - Data warehouse extraction guidelines
  - Production run procedure






















# System conversion

## Conceptual design



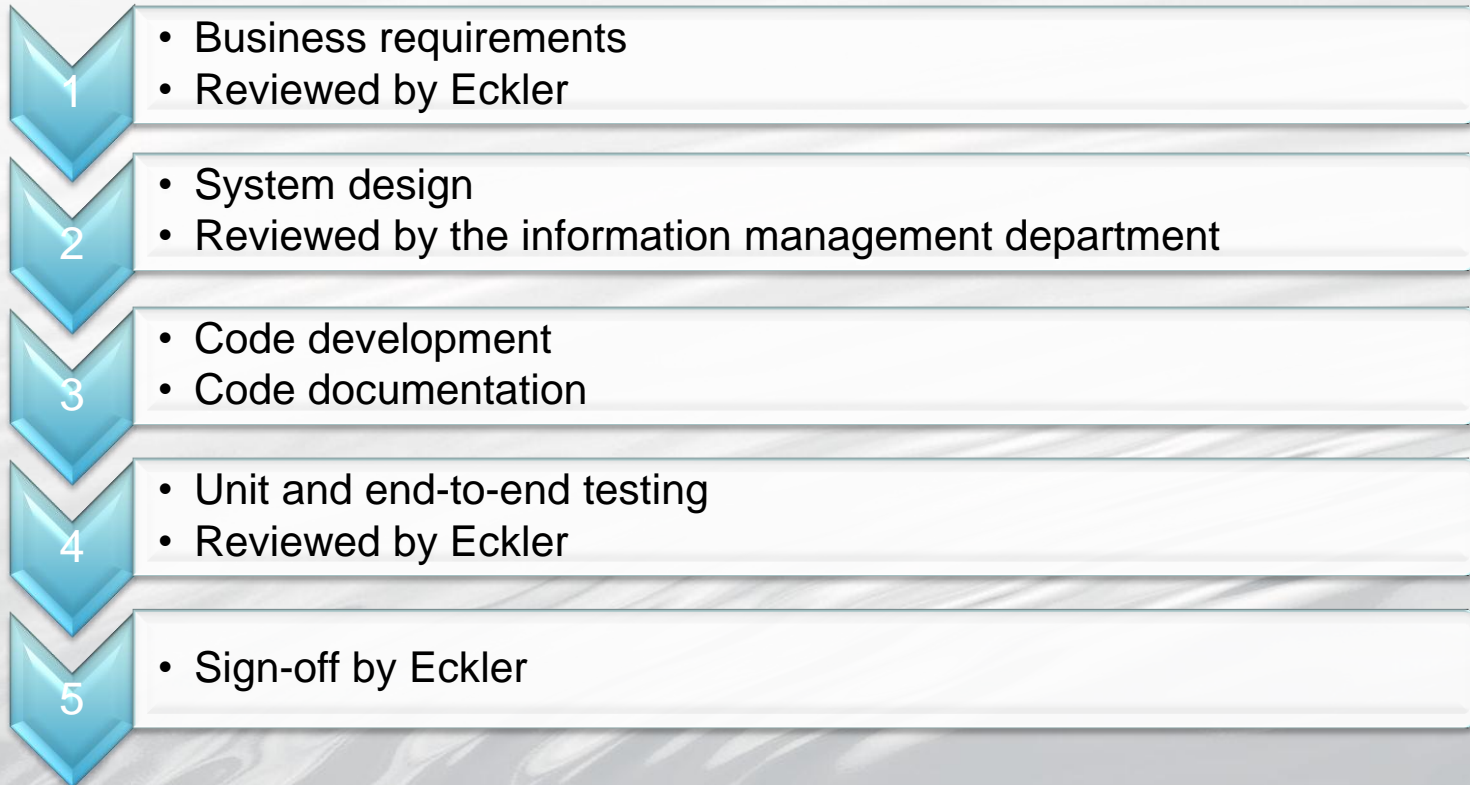
# System conversion

## Implementation approach

	Delivers Broad Based Benefits Early	Targets Priority Calculations (one with greatest impact to business)	Avoids Data Store Redesign	Avoids Code Redesign	Validates Selected Tools Early	Addresses Eckler's Concerns
<b>Big Bang</b> Design and implement the database and every calculation all at one time			 Full design up front	 Full design/test up front		
<b>Data</b> Design & implement the database first, then convert each calculation one at a time	 Data store used by all		 Designed up front for all	 Coding using stable DB design		 Broad based benefits
<b>Calculation</b> Convert each calculation one at a time, then design & implement the database	 Only benefits of that calc	 i.e. Valuation	 By end of calc's, will know 100% the DB reqmts		 "Tests" both programming and data tools	 Broad based benefits
<b>Work Package/Benefit</b> Convert each calculation one at a time, along with associated data elements	 Only benefits of that calc	 i.e. Valuation			 "Tests" both programming and data tools	 Broad based benefits
<b>Hybrid</b> Same as Work Package/Benefit, but with option to design and implement the full database sooner, then the remainder of the calculations	 Option to do entire DB earlier	 i.e. Valuation	 Option to do entire DB earlier	 For those done AFTER DB design	 "Tests" both programming and data tools	 Broad based benefits

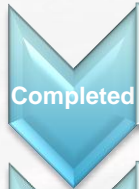
# System conversion

## Development cycle for valuations

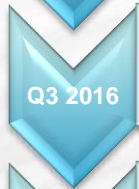


# System conversion

## The roadmap



- Valuations of liabilities with accidents before the valuation date
- A 5-year project



- The valuation of liabilities for latent occupational diseases



- Financial forecast, budgeting



- Rate making



- Future maintenance

# Financial aspects

- External costs
  - Consulting fees
  - Licensing fees
  - Well under \$500 K
- Internal costs
  - Staffing



# Lessons learned

- Not a one-to-one conversion
  - Fresh look at the valuation from the ground up
  - Evaluate methods for data extraction and validation
  - Minimize manual process
  - Incorporate desirable flexibilities
- Design, design and design
  - Take the time upfront
  - Module encapsulation
- Documentation
  - Actuarial
  - Development
  - User manual

# Challenges

- Recruitment
  - Analysts with both actuarial and programming background.
  - Business knowledge and system architect experience
- Resource management
  - Project management and planning
  - Regular actuarial work
- Transition period
  - Parallel using the old way
  - Integration of results from two systems.
- Change management
  - buy-in from others and training

# The ultimate goal

- Better control
  - Quality results
  - Audits / reviews
- Business continuity
  - Minimize key person risk
- Productivity
  - Focus on actuarial work
  - Quick and reliable analyses

# The ultimate goal

## The dilemma

- Data and actuarial methods won't become simpler.
- Actuaries focus on actuarial work, **BUT**
- must still be involved in programming.
- Actuaries  $\neq$  programmers

# The ultimate goal

Minimize the effort on programming

*The way you get programmer productivity is by eliminating lines of code you have to write. The line of code that's the fastest to write, that never breaks, that doesn't need maintenance, is the line you never had to write.*

*Steve Jobs*

# The ultimate goal

## A solution

- The idea of reusable parts
  - Standard actuarial calculations can be “black boxed”.
  - Actuaries choose the right parts and focus on assembling
  - Customization can be done through
    - Configuration, or
    - Inheritance of the existing modules
- To have a library of parts eventually

# The ultimate goal

## Examples of increased productivity at WCBA

- Comparison of actual with expected experience
  - Before: three person days
  - Now: one person hour
- Claims development table
  - Before: three person days
  - Now: one person hour
- Tabulation of cash flows for ALM
  - Before: one person day
  - Now: one person hour
  - In 2017: one person minute

**Questions / Comments?**