

- Crossing the Chasm - Simplifying Data Management with Perl and Metadata

d-Wise Technologies

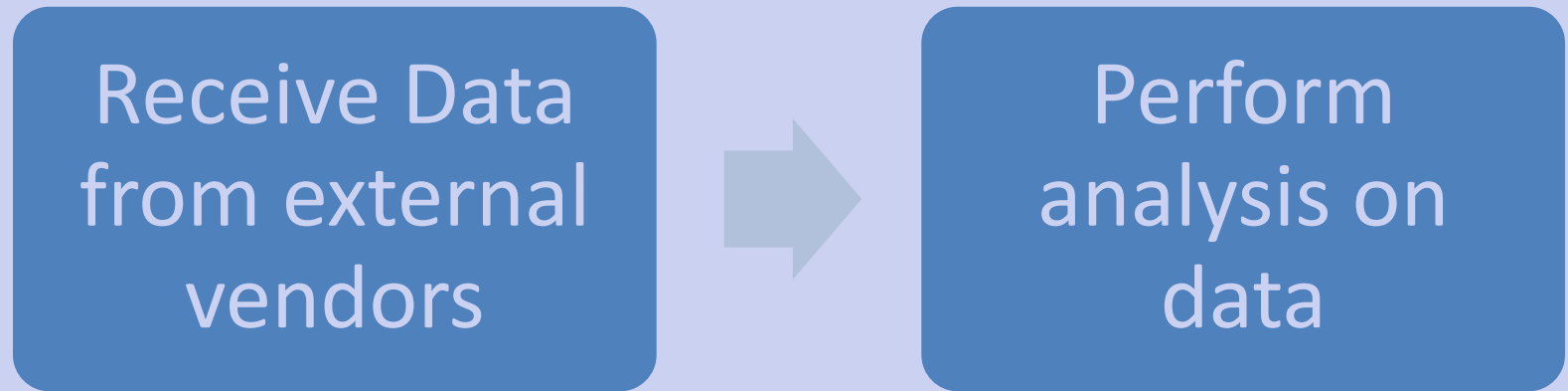
Stephen Baker

Manager, Strategy and Business Development

Overview

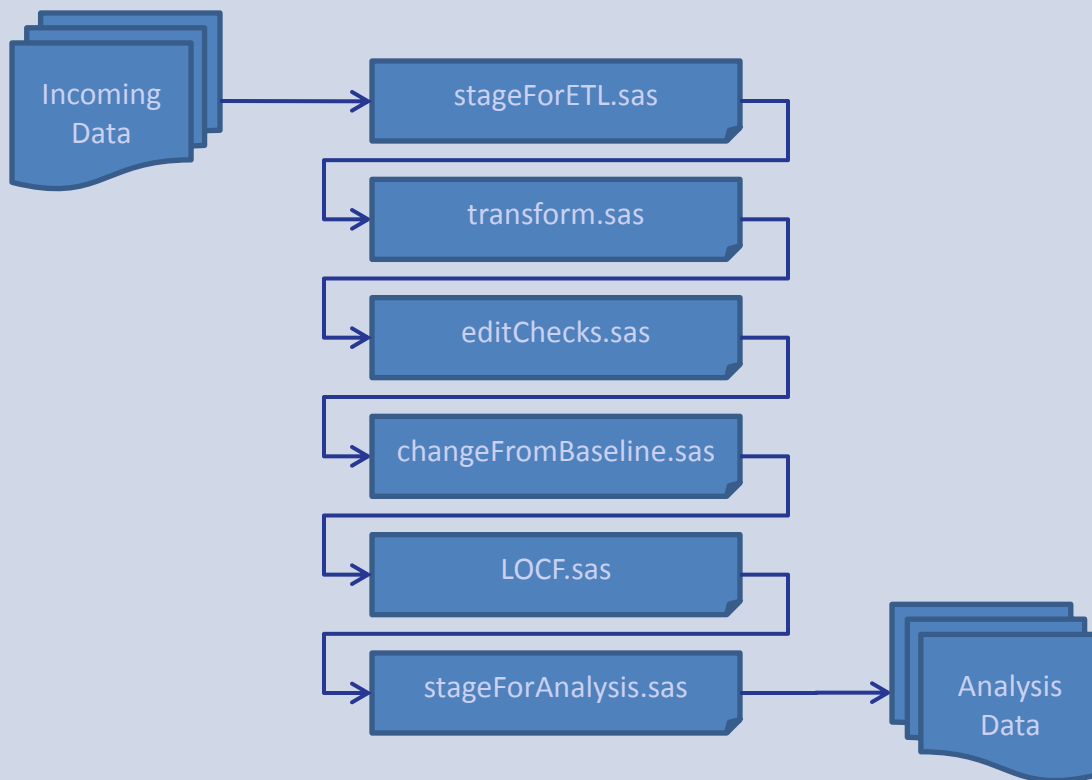
- Defining the chasm: The flow of data through the clinical organization
- Examining a framework
- Technology Considerations
- Summary

Defining the chasm

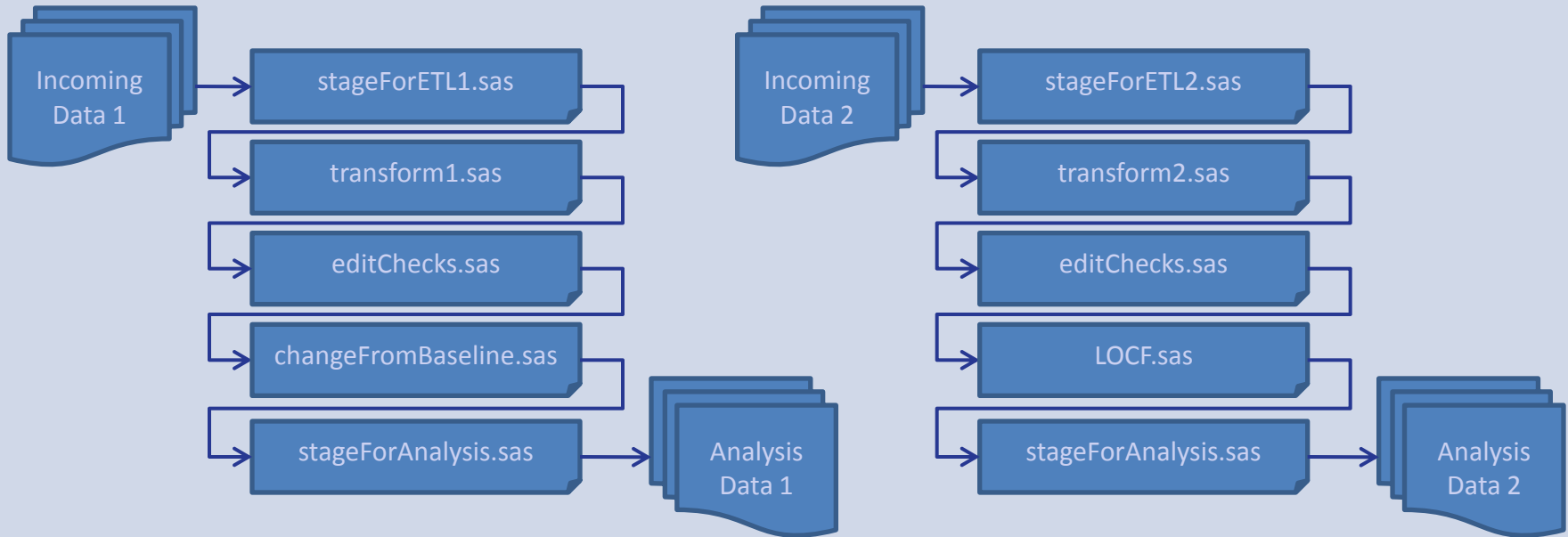


Let's examine how this evolves over time...

Data Moving Through the Business

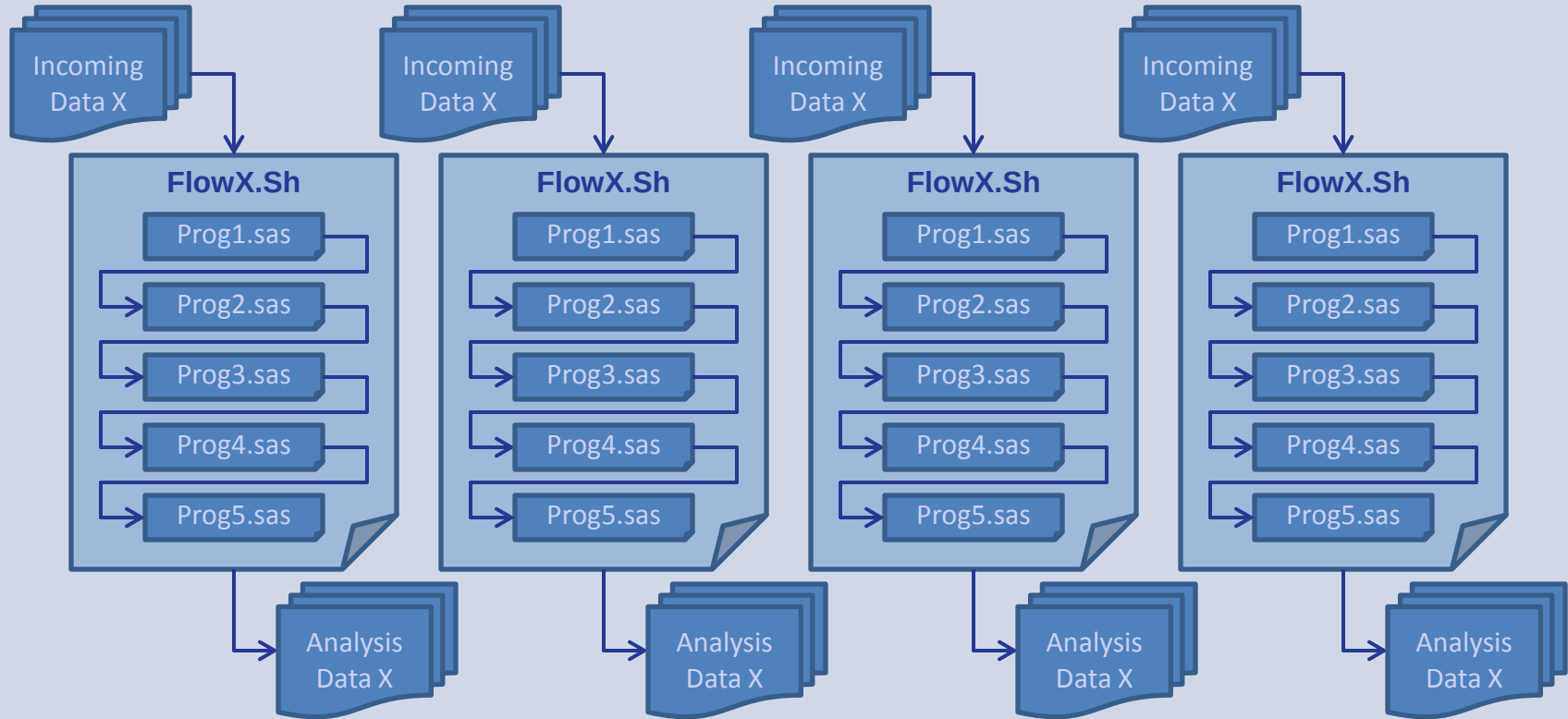


The Chasm Emerges

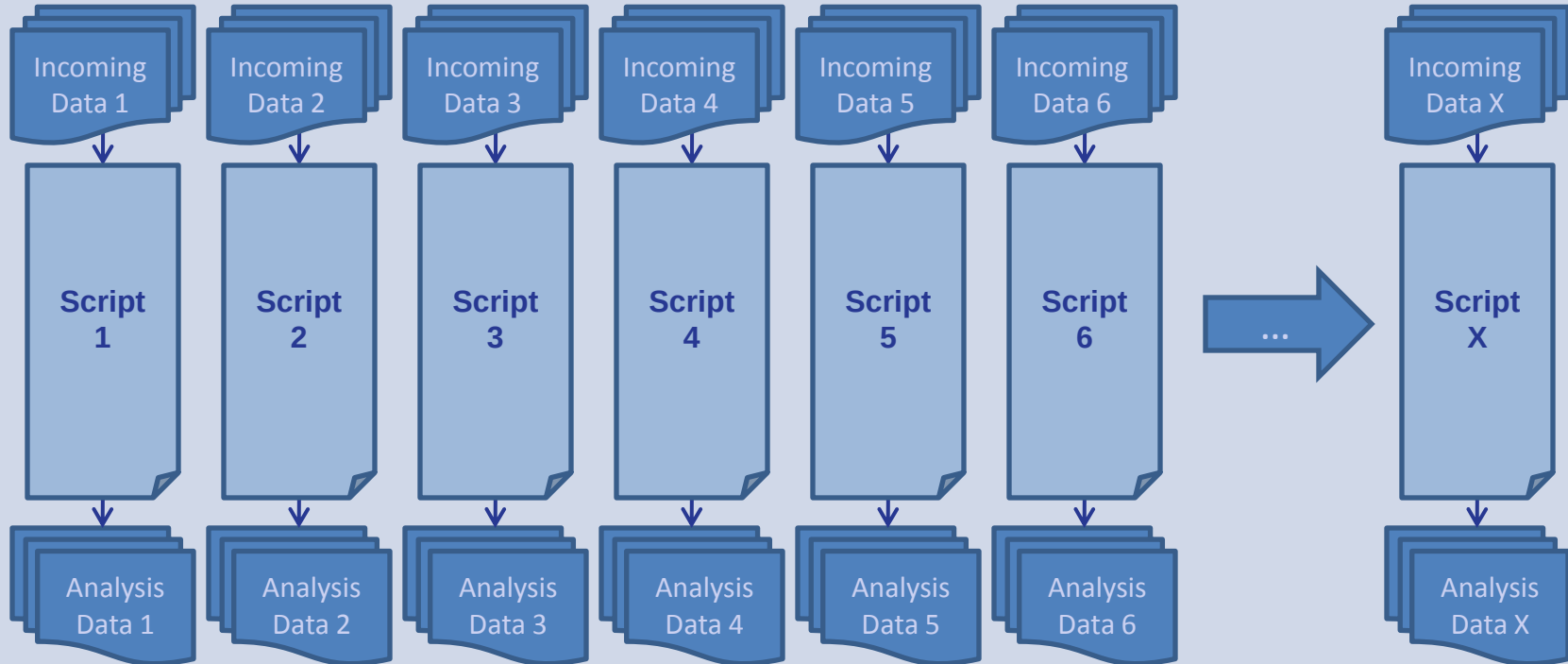


Suddenly, the workflows are different!

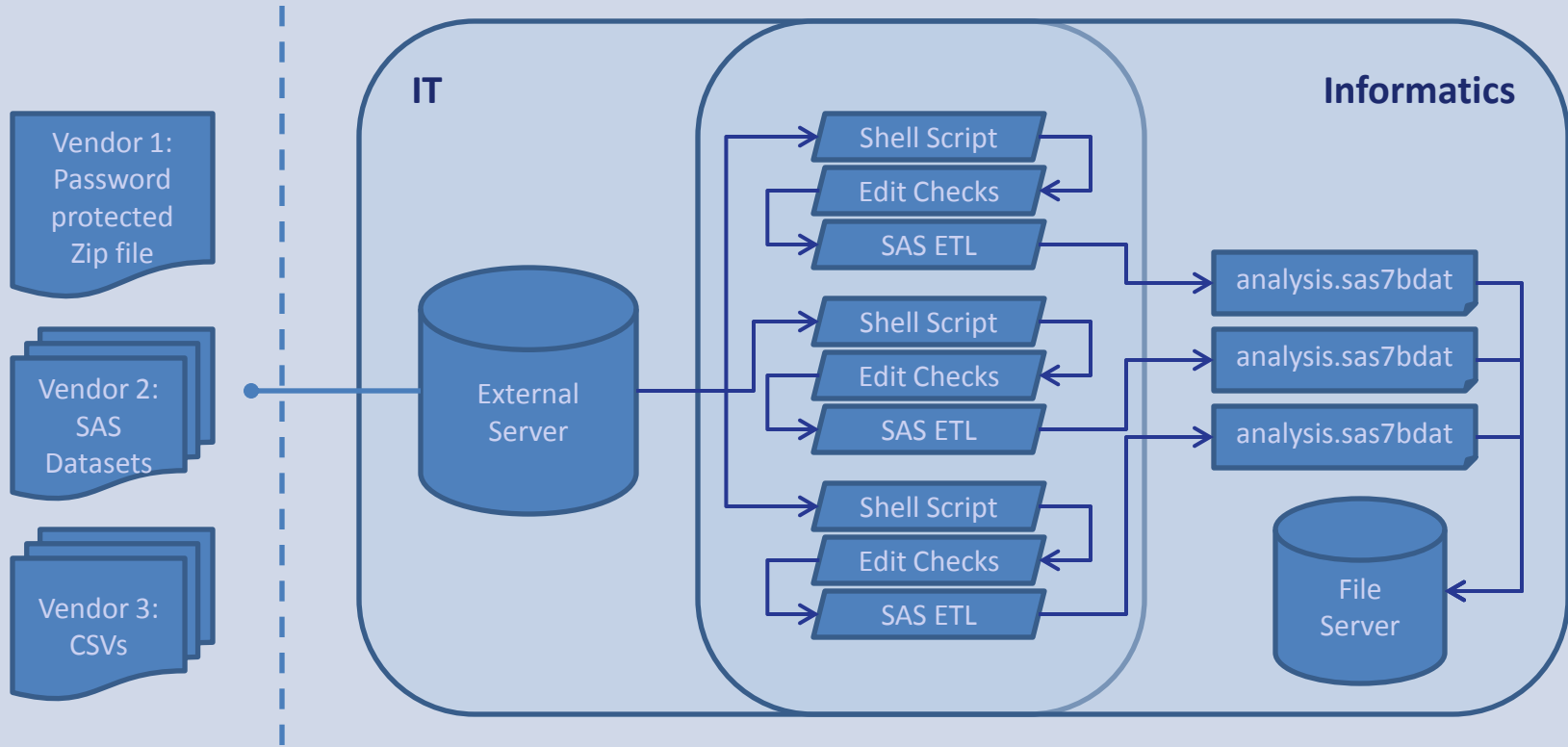
The Chasm Introduces Chaos



Chaos Becomes Unmanageable



Adding Organizational Complexity



Crossing the chasm

Receive Data from
external vendors



Perform analysis
on data

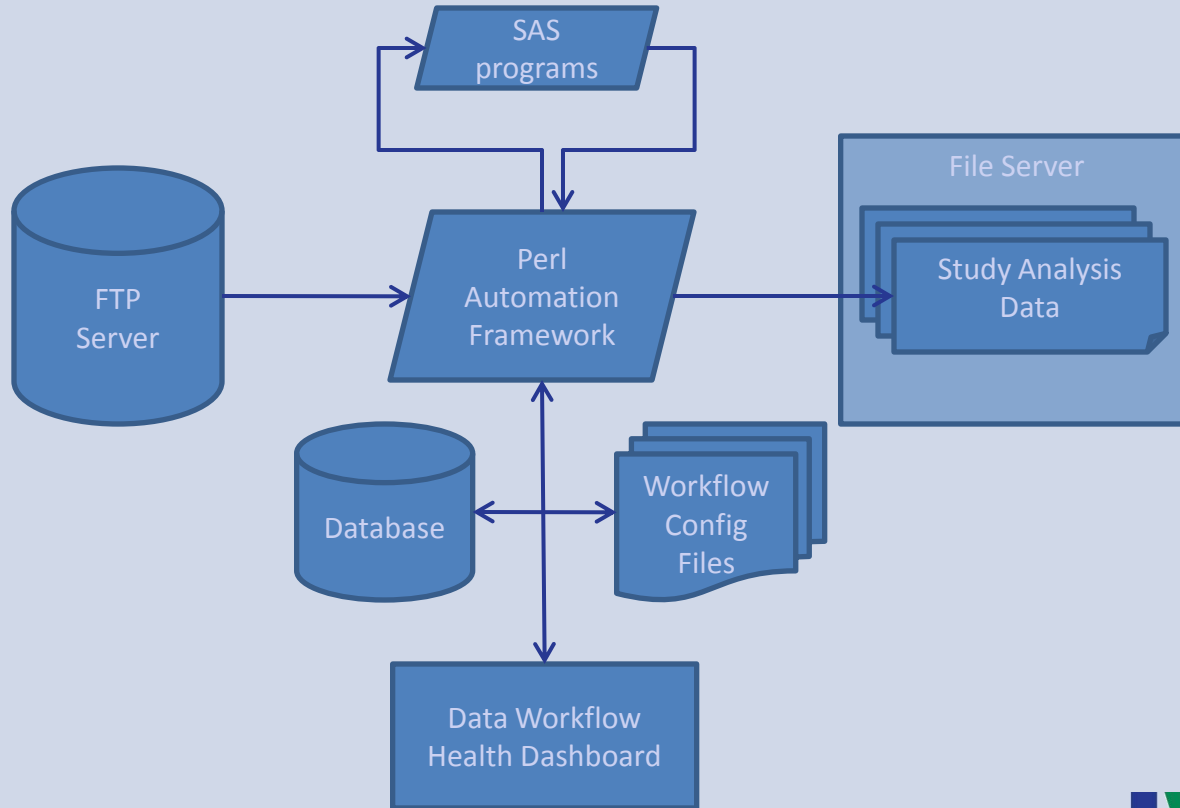
- Every clinical organization receives data from vendors in multiple formats and has to move it through the business process to enable analysis to happen
- What would a system for managing this process look like?
 - Separation of programming logic and configuration information
 - Repeatable tasks, 'actions', should be defined and used to build all workflows
 - Notification needs such as emailing data consumers and updating a dashboard should be supported

Components of a Framework

The system must...

- support varying input data formats from CROs
- externalize configuration details such as user credentials and file server paths
- be flexible to changes in the infrastructure and portable across operating systems
- be easily extensible as the business places new demands on the framework
- support varying data preparation activities
- provide a reusable library of actions from which to build work flows
- support extended the library easily and surfacing new features to users
- provide a simple interface for the users to define new flows or modify existing flows
- support all staff having visibility into the health of the data flows
- provide a “single source of the truth” for understanding how individual data flows are defined

A Framework Emerges



What Role Does Metadata Play?

- Vastly overloaded term in this industry – let's talk about metadata as configuration information to drive framework 'action' building blocks
- Metadata at the **Action level**
 - Location of a file to operate on
 - Password required to unzip an archive
- Metadata at the **Workflow Level**
 - Trigger mechanism that starts this workflow
 - Conditional events that should cause a workflow to abort
- Metadata at the **System Level**
 - Notification details – who to notify, when, and how?
 - IT configuration – decouple application from infrastructure
- Goals
 - Proper tradeoff between automation and flexibility
 - Separate programming logic from configuration logic

Workflows Actions

- Obvious activities, such as copying files or unzipping files, are quickly identified and easily supported.
- More ambiguous tasks, such as specialized ETL or statistical methods, might be tougher to compartmentalize.
- In the reference system, actions defined included:
 - Unzip
 - Copy
 - run a SAS program
 - run a Perl program
 - Move
 - Delete
 - search through a text file
 - search through a log file
- Each action implements an interface, making the automation component easily extensible.

`setProperties(pathToConfigFile, notifyEmailAddresses)`

`execute(ActionError)`

Workflow Lifecycle

- The system reads a top level workflow registry file to learn about all defined workflows
- Each workflow is defined as
 - a trigger event, such as a file appearing in a folder or a time to execute a certain process
 - a sequence of actions
 - notification requirements
- If the trigger event is satisfied, the remaining actions are executed in order
- If a failure is noticed while processing in any intermediate action, the workflow as a whole fails
- System events (starting a workflow, trigger satisfied, action running, error occurred, workflow completed) are logged to the database

Single Source of the Truth?

- Goals
 - To be able to definitively understand what activities are required to produce a given data
 - To be able to see the overall health of the data workflows
 - To perform rudimentary impact analysis for ETL-like actions
- In the 'chaos' example... multiple, ad-hoc programs - difficult to manage, impossible to see comprehensively
- Good... metadata driven actions defined in user maintained text files
- Better... A UI for viewing/editing workflow configurations in a database
- Best... A Workflow Health Dashboard surfacing system details from the database
- Ideal... The information surfaced by the system enables the business to focus on the sciences rather than the supporting tools

What is Perl Good For?

- <free>Open Source</free>, easy to install, portable across operating systems
- Terse syntax keeps programs short and human readable
- Powerful scripting language
- Rapid Development Features:
 - Exceptionally powerful for processing text files
 - Robust Support for OO design, exception handling, regular expressions
 - Easy extension via Perl modules available from CPAN
 - Log4perl
 - Turn-key integration for database, email, rolling file appender
 - Log Levels (debug, info, warn, error, fatal) make it easy to tweak to verbosity of the system without changing code
 - Automated IQ/OQ
 - Unit Testing Frameworks

Picking the Right Technology for the Job

- SAS
 - SAS is a **superior** technology for manipulation, viewing, and analysis of tabular data
 - disappointing lack of programmable exception handling, testing frameworks
 - SAS datasets are a part of the clinical industry
- Perl
 - **superior** technology for scripting and processing text files
 - open source community provides a vast array of usable tools enabling rapid development
- Database
 - Relational database, either COTS or open source, enable web applications to be quickly built to interface with the system and help control the user experience
- Where role does open source play in this industry?
 - Cost of adoption is more than just cost of licenses
 - Vendor lock-in can be a challenge to break free from over time

Summary

- Ad-hoc approaches evolve from desirable, to complicated, to unmanageable, to painful – and costs increase at each step along the way
- Framework approaches encourage reusability and extend capability/accountability to broader audiences
- Separate business logic from programming logic
- Abstraction and metadata driven approaches enable the business
- Custom solutions and open source are viable options when considering how to apply technology to business problems

Questions?

*When you carry around a hammer all day,
you find everything starts to look like a nail.*