Fall 2009

Strategic Tool Building in Data Engineering: Using the Systems Engineering Process to Meet Strategic Business Goals

Amanda Gerdes
Loyola Marymount University

Strategic Tool Building In Data Engineering

Using The Systems Engineering Process To Meet Strategic Business Goals

Amanda Gerdes

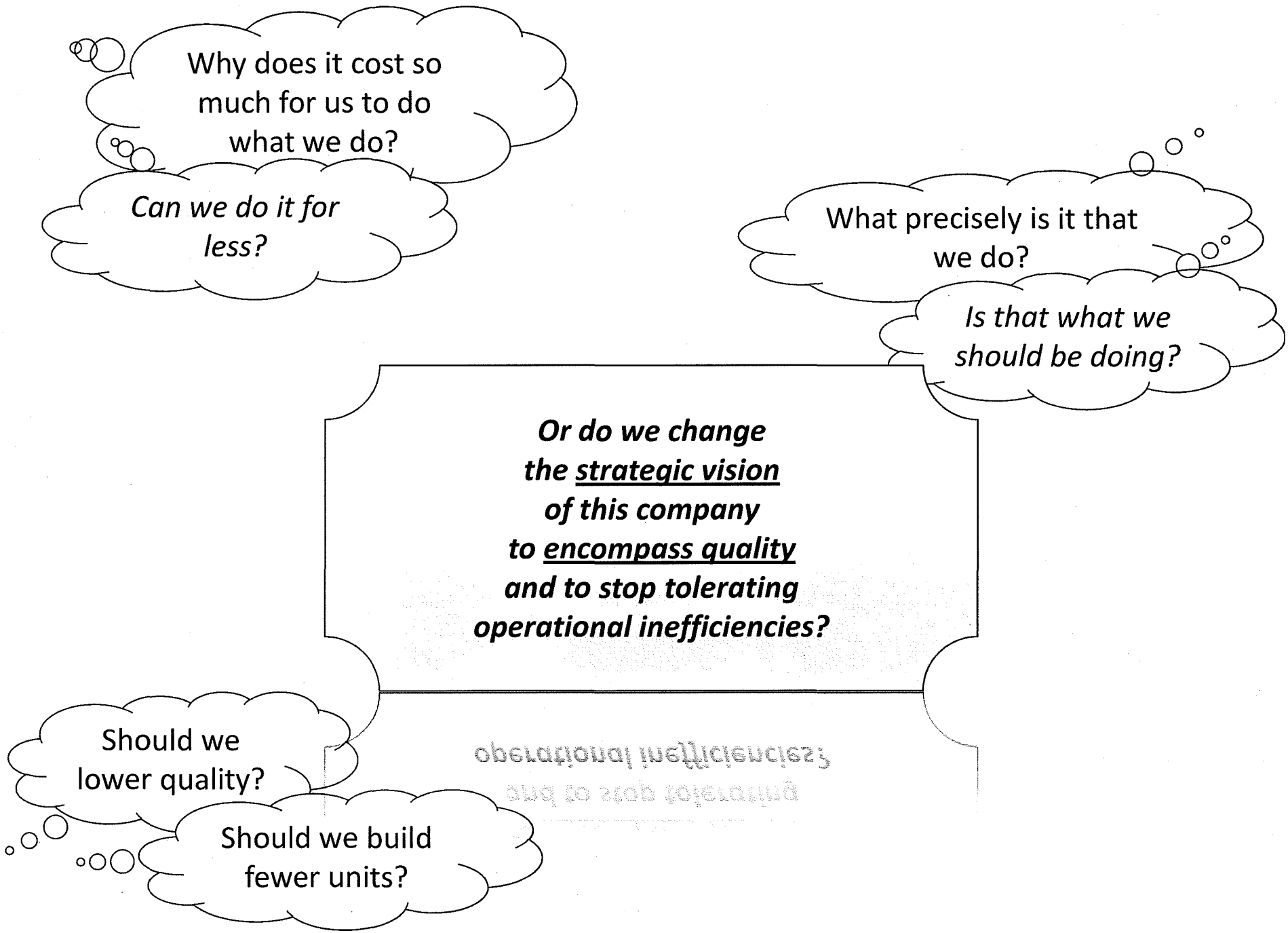
Systems Engineering Integrative Project

Loyola Marymount University

14 December 2009

The Company, The Business, and Data Engineering

Introduction

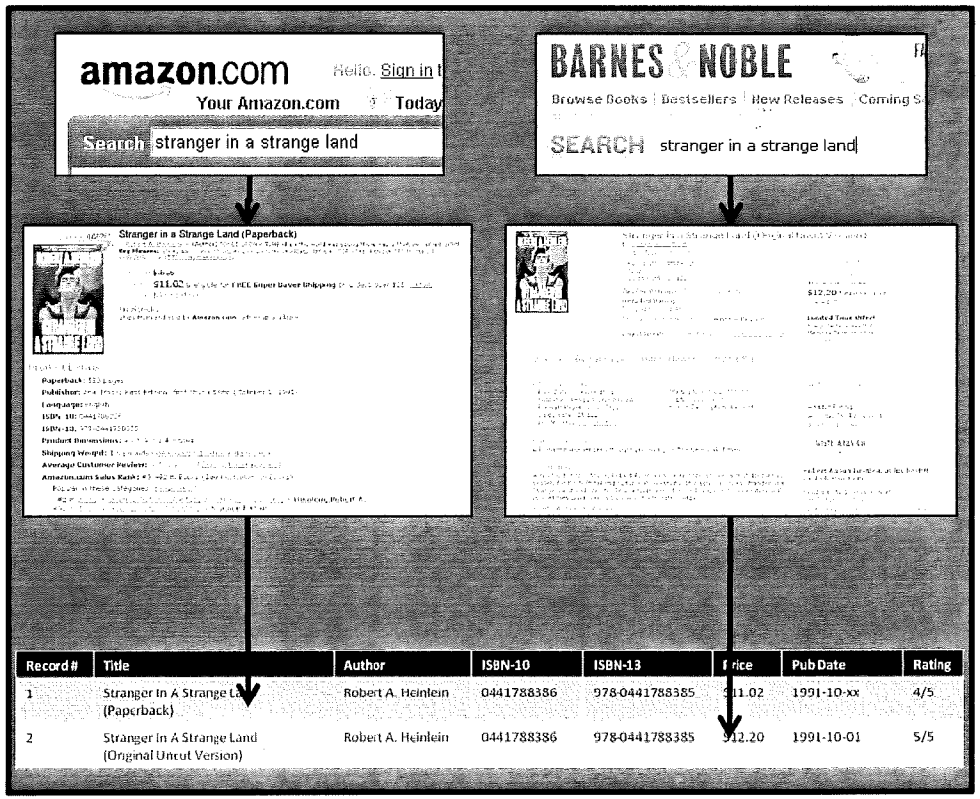

Why does it cost so much for us to do what we do?
Can we do it for less?
What precisely is it that we do?
Is that what we should be doing?
Or do we change the strategic vision of this company to encompass quality and to stop tolerating operational inefficiencies?
Should we lower quality?
Should we build fewer units?

About The Company

- History
  - Founded in 1999
  - Strong ties to academia, strong focus on research
  - Primarily government contracts (DARPA, NSF, Air Force)
  - Expansion into commercial sector in 2002
- Structure
  - 40 employees
  - Three primary functional groups
    - Labs (Research)
    - Software Engineering (Platform Development)
    - Data Engineering (Application Development)

About the Business

- Data Extraction, Aggregation, Normalization
- Business Models
  - Software Licensing Model
  - Design Consultancy Model
  - Data Delivery (Hosted Solution) Model
- Implementation Models
  - Scheduled Batch Scrapes
  - Runtime Scrapes
- Sample Implementations
  - Background Search/Risk Management: Runtime Data Delivery
  - Competitive Analysis: Scheduled Batch Data Delivery
  - Events Aggregation: Scheduled Batch Design Consultancy



About Data Engineering

- Role Within The Company
  - Accept conceptual guidance from Labs
  - Use platform software from Software Engineering
  - Develop customer applications
- Roles and Responsibilities
  - Requirements collection and analysis
  - Software and data architecture
  - Application design and build-out
  - Training, Technical Support
- Heterogeneous Makeup
  - Software engineers
  - Software analysts
  - Data analysis
  - Business analysis
  - Offshore resources

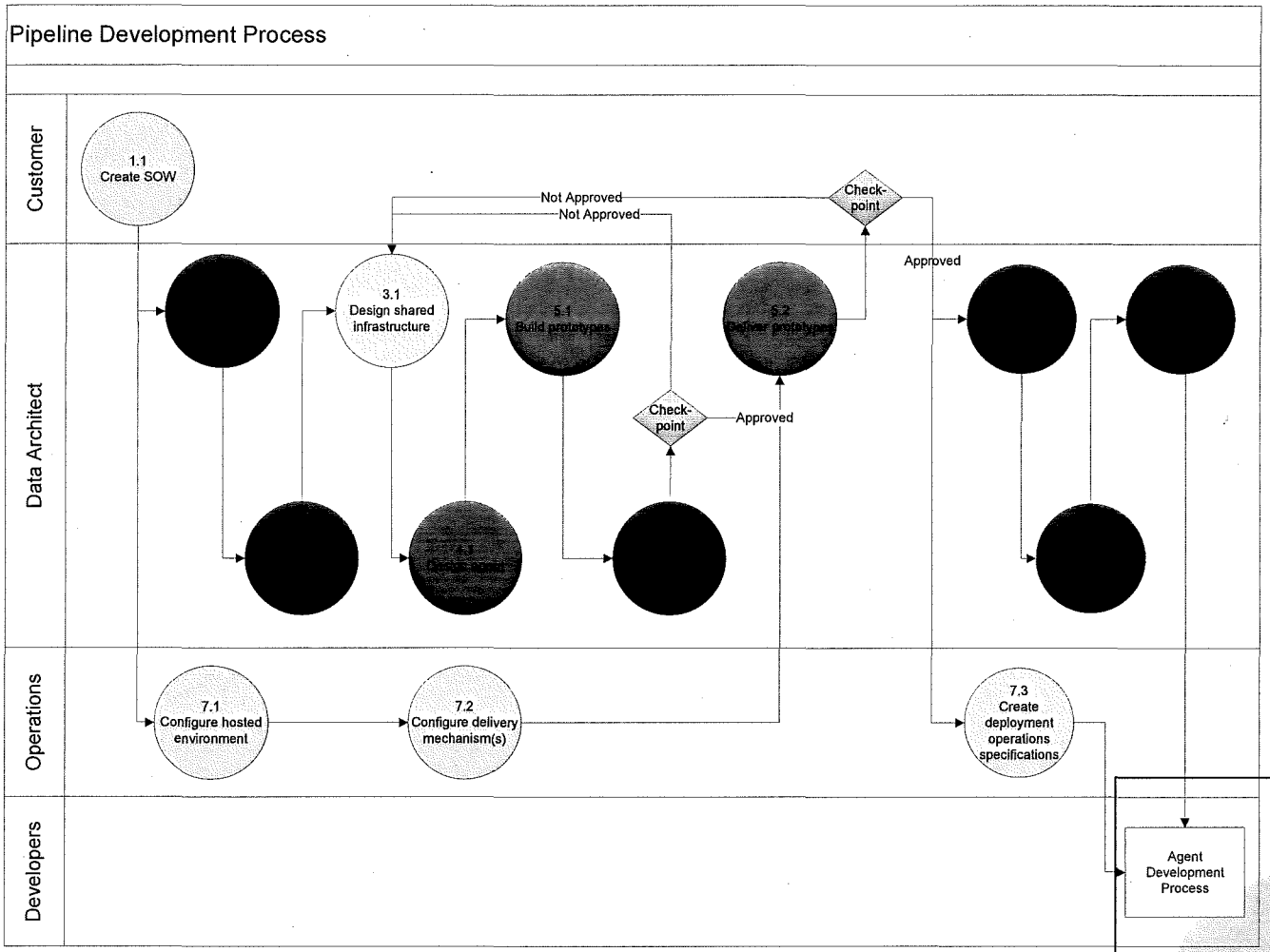
Data Pipelines, Development Processes, Total Cost of Ownership

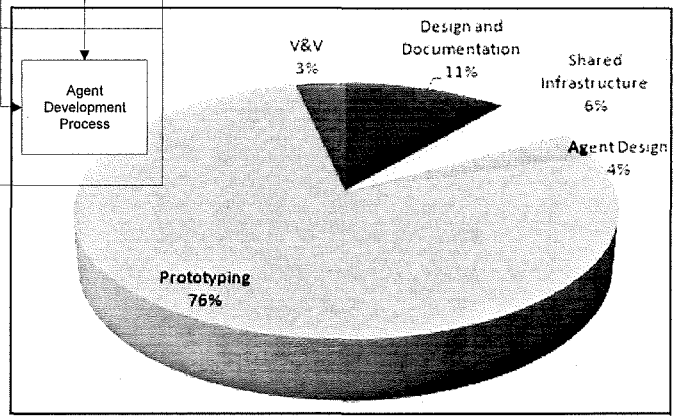
The Current State

Data Pipelines

- Analogy: Rail transport
- Purpose
  - Provide shared infrastructure
  - Limit variable ("per-agent" costs)
- Manifestation
  - Shared, Generalized Codebase
  - Standards and Specifications
  - Individual, Per-Site Units
- Advantages and Disadvantages
  - Lower per-agent costs
  - Easy addition of new sources over time
  - Higher front-end development costs, time

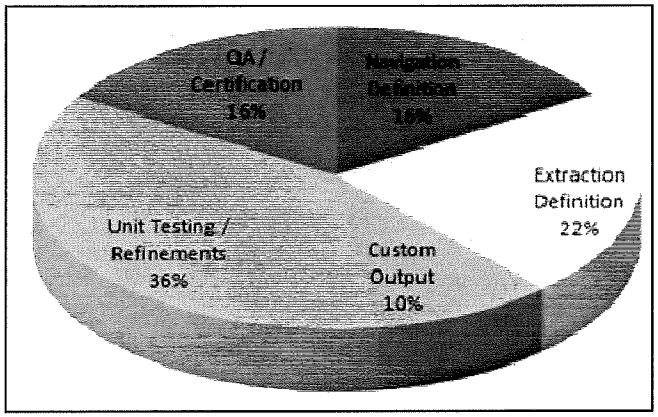
Pipeline Development: Process

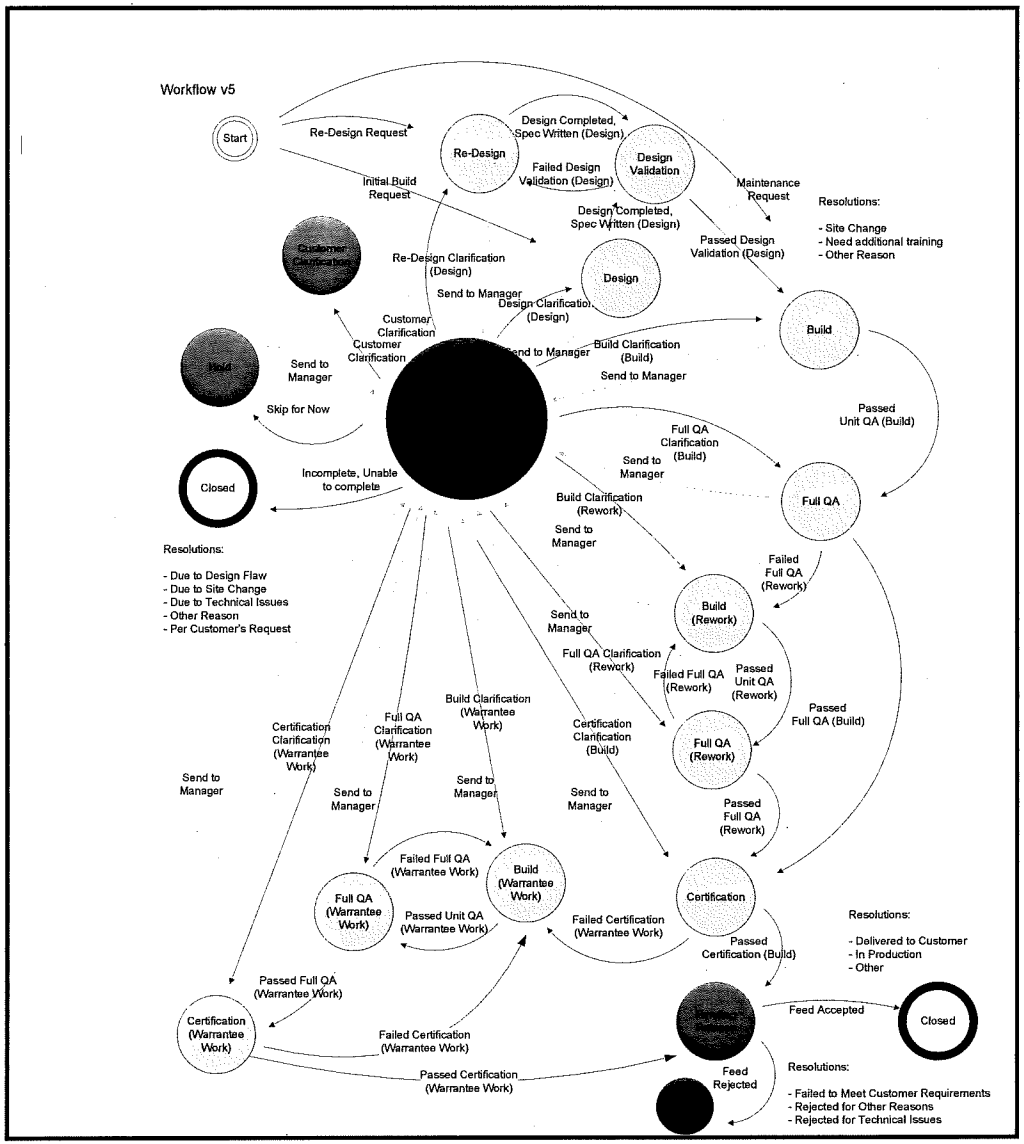




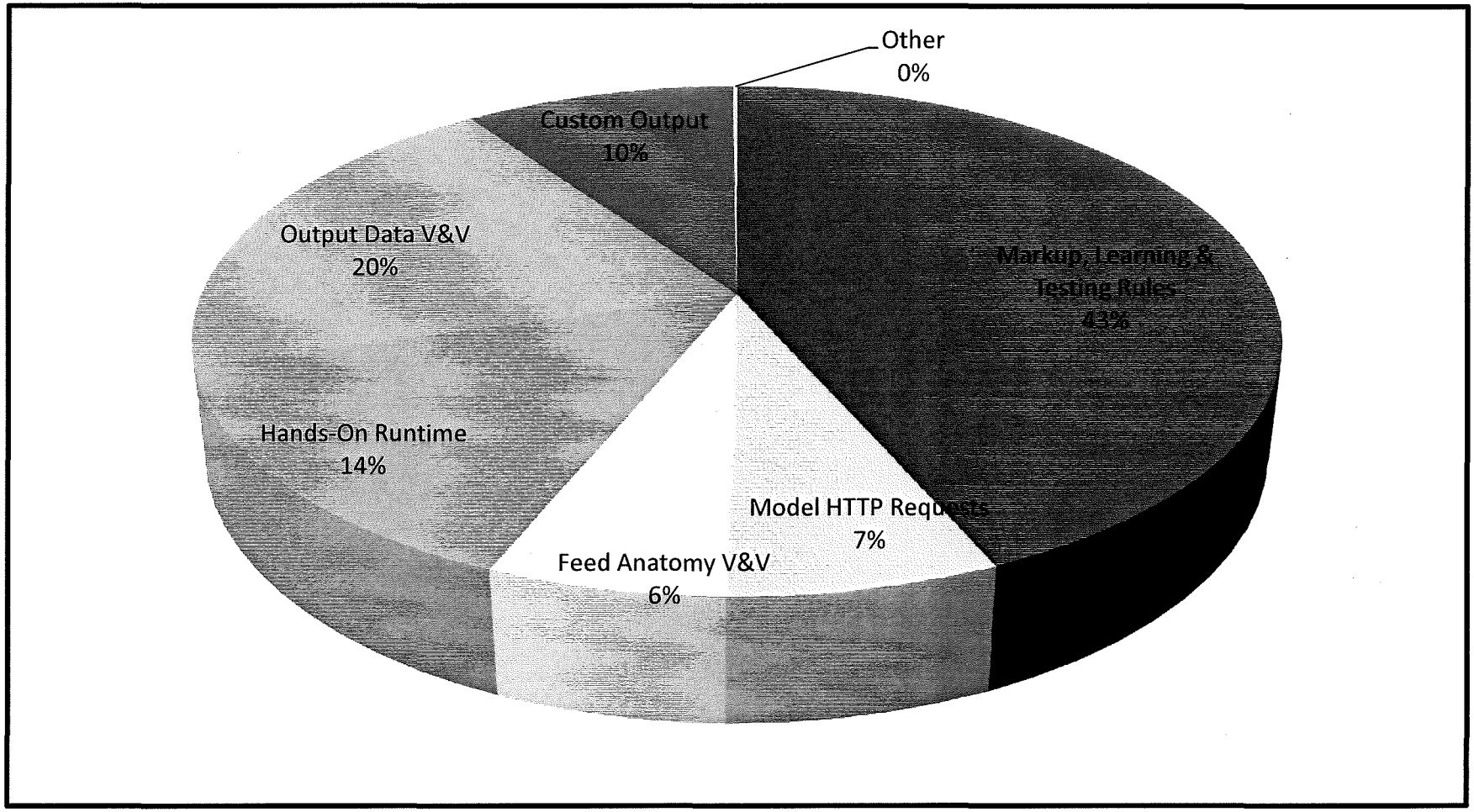
Agent Development: Process

- Build
  - Navigation Definition
  - Extraction Definition
  - Custom Output Mods
- Unit Testing and Refinements.
- QA / Certification.





Development Effort By Task



Pipeline Costs

- Per-Pipeline Costs
  - Typically fixed
  - Amortized over the lifetime of the deployment
  - Cost varies by complexity of the pipeline or variability within the entire data set
- Per-Agent Costs
  - Variable costs
  - Includes development, maintenance, and ongoing operations
  - Cost varies by both the complexity of the site population as well as the total number of agents to be produced

Strategic Alignment, Strategic Trade-Offs, and the Vision

The Future State

Strategic Trade-Offs

Area	Advantages	Disadvantages
Lowered Quality	• Immediate, short-term impact • No implementation costs	• Compromised reputation in marketplace • No long-term cost benefits • Maximum cost benefits are firmly bounded
Decreased Volume of New Sales	• Immediate, short-term impact • No implementation costs • Lowered acquisition costs	• Limited opportunities for growth or expansion • Introduces "all eggs in one basket" risks
Postponing Competing Strategic Initiatives	• Lower future costs • Greater operational efficiency • Maintains quality levels and market reputation • Lays foundation for scalable growth	• High implementation costs • High opportunity costs • Effects are seen over the medium- to long-term

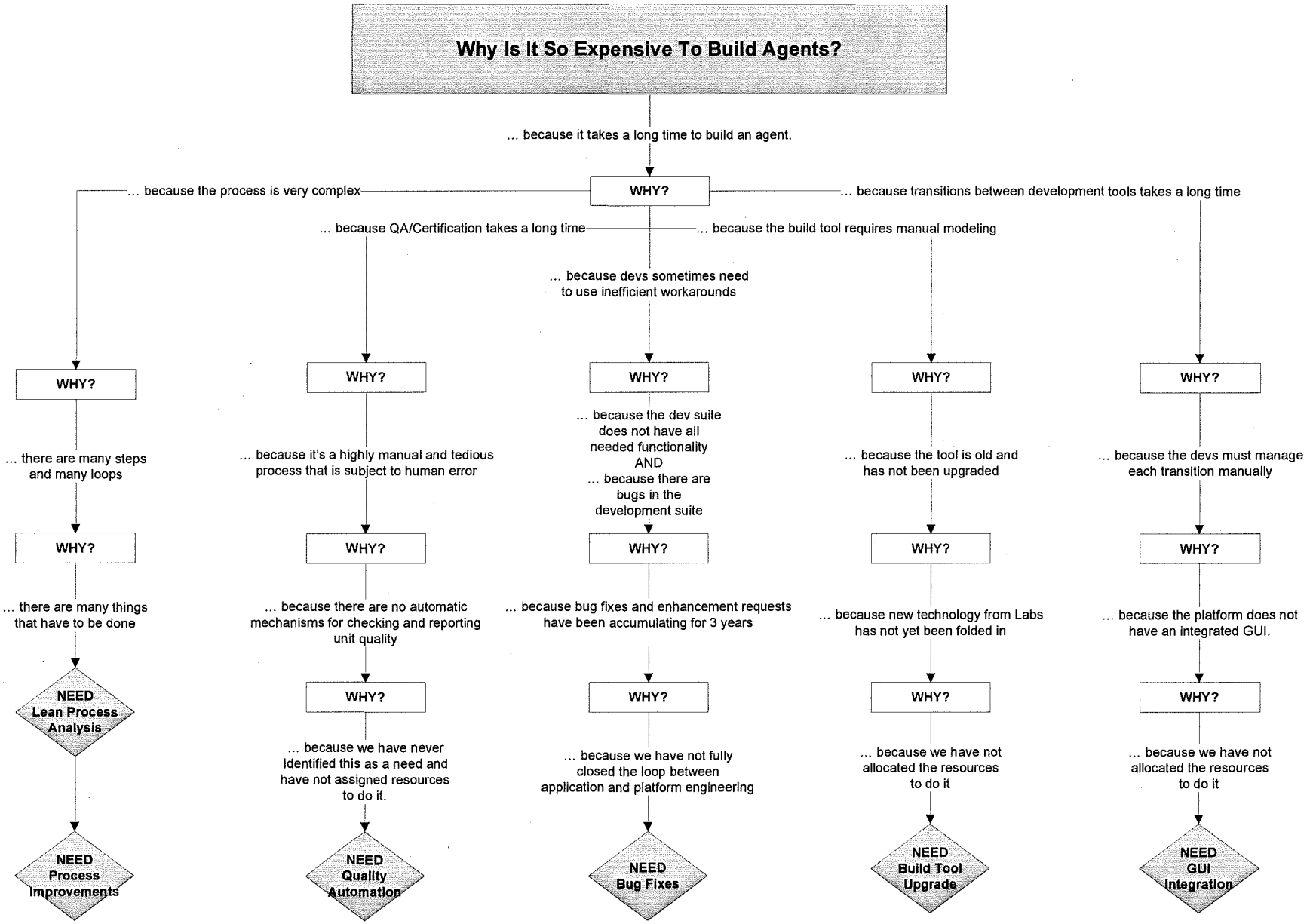
The Vision

- Difficult Decision:
  - Redefine the strategic vision for the company
    - Temporarily putting growth areas and product line expansions on the back-burner
    - Making the lowering of costs *through increased operational efficiencies* a company-wide strategic goal
- Manifestation in Data Engineering
  - Identify operational inefficiencies in Data Engineering
  - Develop and implement plan for becoming efficient, self-drivin

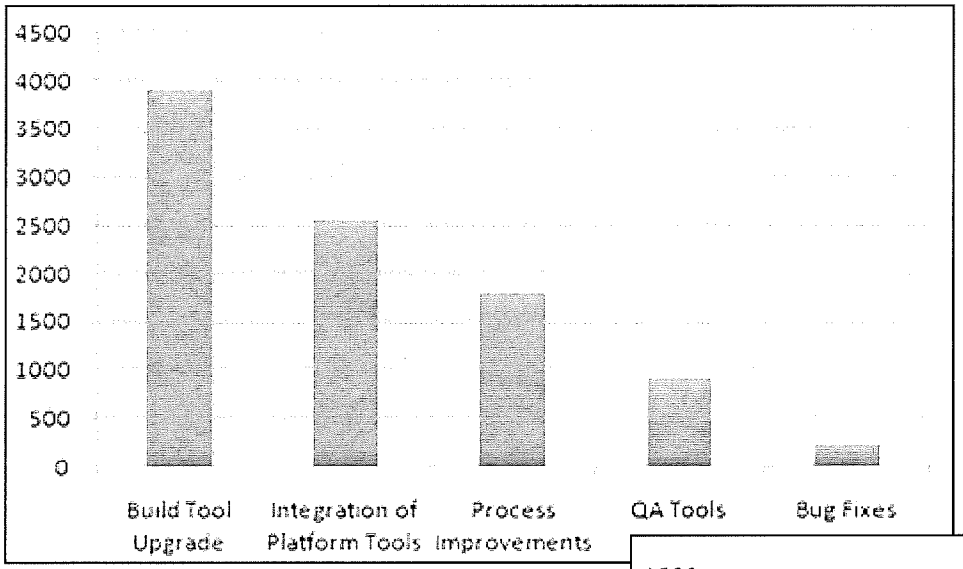
Five Whys, Pareto Analysis, Process Analysis

Identifying The Problem

Root Cause Analysis: The Five Whys

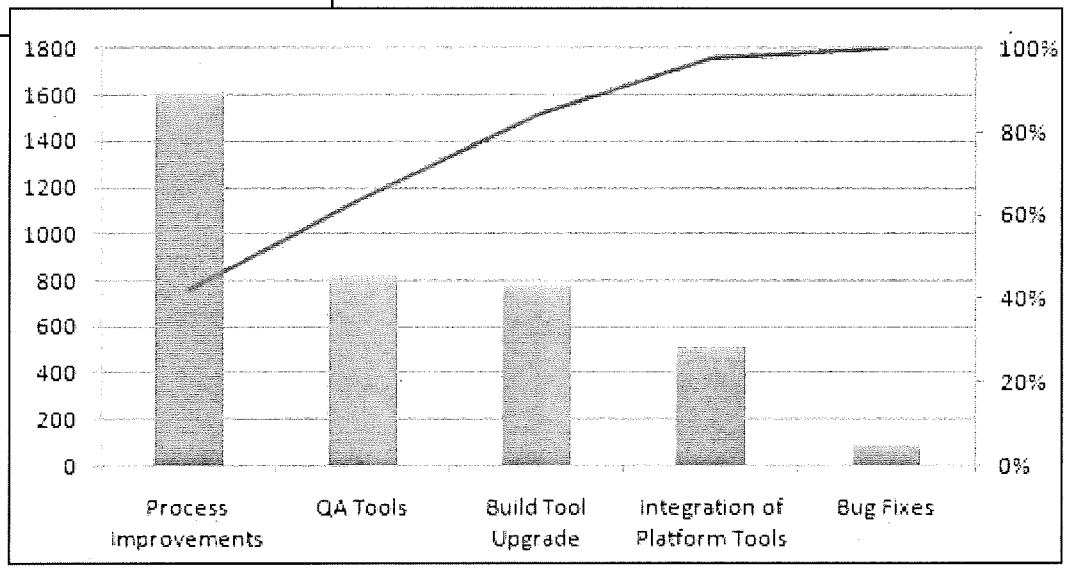


Root Cause Analysis: Pareto Analysis



Estimated Cost Reductions Per Improvement Area

Improvement Areas By Cost-Effectiveness



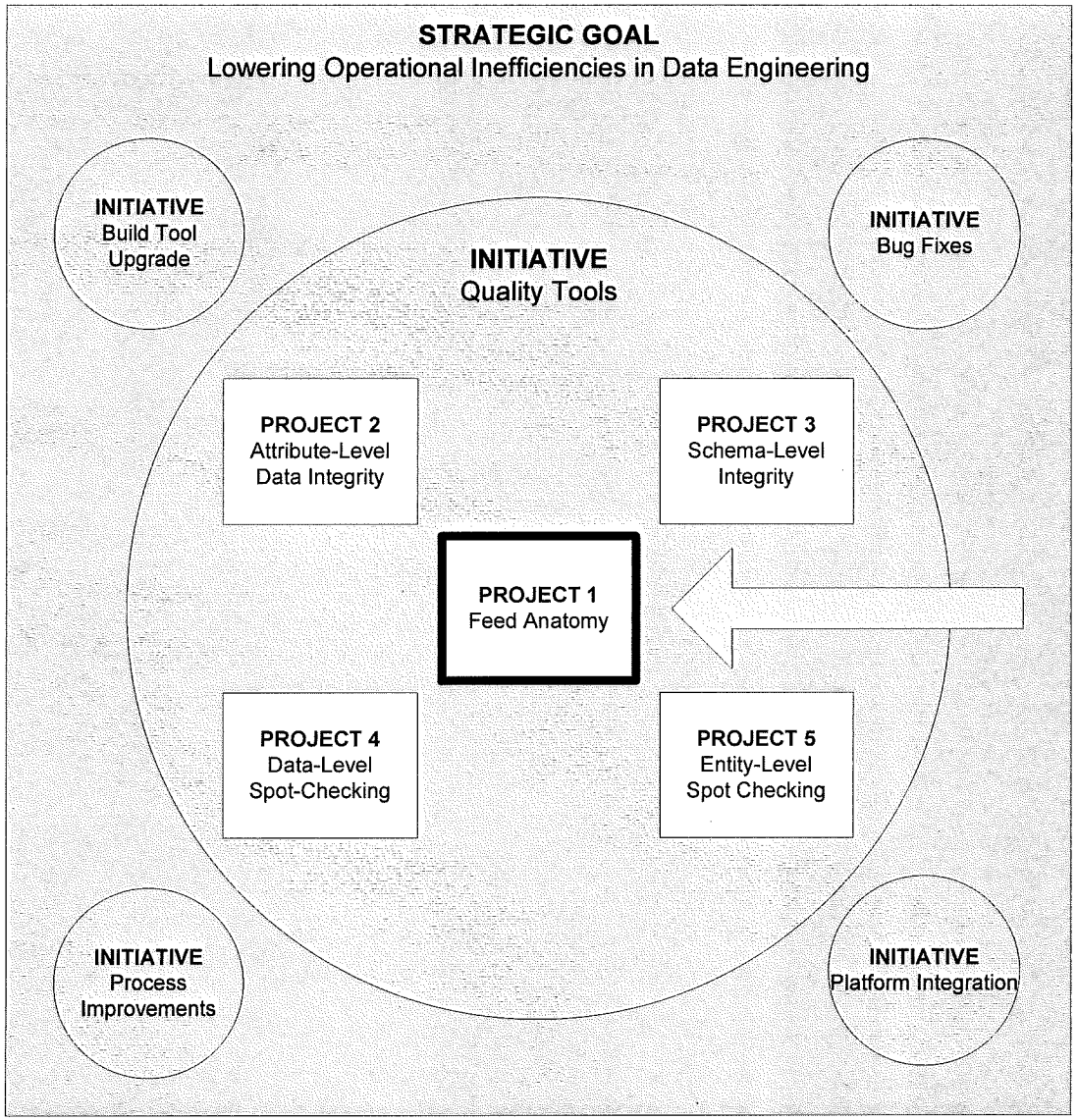
Process Analysis: Identifying Waste

Waste Area	Prevalence	Description
Over-Production	High	Example: Documentation
Inventory	Medium	"Soft" inventory – that is, electronic artifacts – is an issue. (Example: Documentation, archives.)
Transportation	High	Transportation is a significant issue in the agent development process. (Example: Toolset interfaces.)
Unnecessary Movement	High	Physical movement is not an issue. However, there is a huge amount of unnecessary movement in terms of system configuration and disparate systems access. (Example: Disparate storage, access environments.)
Waiting	High	While sibling processes have been successfully parallelized, individual processes themselves remain highly linear and highly sequential.
Defective Outputs	High	The rate of defective outputs is fairly low but the relative severity of each is increased by the fact that they often are not detected until too far downstream.
Over-Processing	Medium	Significant improvement in past 6-12 months as pipeline development processes have matured and decreased the amount of per-unit processing that must be done. However, there remains a risk that in order to generalize a pipeline, too much variability is accounted for, creating a pipeline-level over-processing issue.
Insufficient Design (Womack)	High	External customer needs are typically well captured and well accounted for. However, the Data Engineering group performs their work using an outdated set of tools that is currently insufficient to meet their needs in terms of application development.
Unnecessary Space (Womack)	Low	Space is not typically an issue.
Negative Emotions (Oppenheim)	Medium	THE COMPANY's workforce tends to be highly motivated, highly compatible, and united under a very strong corporate culture. However, financial complications in the company has manifested as lay-offs and salary cuts. This has created an anxious environment, commonly manifesting in negative emotions such as a lack of motivation and even discontent. These are counterproductive to a lean operation.

Strategic View, Tactical View, Operational (Technical) View

Building The Solution

Strategic View: Developing Operational Efficiency



Tactical View: The Quality Tools Initiative

Area	Advantages	Disadvantages
Formal Quality and CI Program	• Marketing point • Could be used to generate credibility in and for industry	• "Buzz word" • May have/create infrastructure cumbersome to very small company
Develop In-House Program	• Compatible with strong corporate culture • Can be implemented at a grass-roots level • May be used to generate leadership position in industry	• May be resource intensive • Requires in-house expertise

Tactical Trade-Offs

The Plan

Project	Description
Feed Anatomy	• Objective: Create a tool which can validate the structure of all feed objects against a project-specific, configurable "gold standard". • Expectations: Greater overall quality and consistency of feed objects within pipelines; faster verification times; lower rework • Mechanism: Detailed below
Attribute-Level Integrity	• Objective: Create a tool which can validate data at the attribute level, e.g. "Do the values reported in column item_price look like prices?" • Expectations: Faster verification times, possible runtime validation, development of standardized, reusable internal schemas • Mechanism: Reusable type queries, reusable execution framework, reusable reporting framework
Schema-Level Integrity	• Objective: Create a tool which can validate XML output via validation against a well-defined schema • Expectations: Faster verification times, more accurate verification, encapsulation of project requirements in standard format
Entity-Level Spot Checking	• Objective: Create a tool that provides an integrated view of extracted tuple and details page • Expectations: Vast reduction in recall/precision data collection and reporting • Mechanism: Given tuple, load and display corresponding page; given page, load corresponding tuple; specify "correct/not correct" for behind-the-scenes statistics gathering
Web-Based Spot-Checking	• Objective: Create a tool that is capable of facilitating recall metrics collection • Expectations: Vast reduction in recall data collection and reporting • Mechanism: Randomly crawl site, gather random identifiers from target population, verify that those entities appear in the output

Goals, Requirements, Trade Studies, Agile Software Development

The Feed Anatomy Project

Goals

- Goals
  - Build a tool or toolset capable of validating the structure of all feed components with minimal human interaction
  - Capture the project-specific information required to perform this check in a reusable fashion
  - Give all layers of the development process the same tools to use to promote visibility, consistency, and developer empowerment
  - Provide a means for capturing and reporting of errors so that statistical process control methodologies can be applied
- Schedule
  - Original: Completion by December 31, 2009
  - Modified: Completion by January 31, 2010

Requirements: Customer Requirements

- Customer Identification: Internal uses (Data Engineering, on- and off-shore)
- Requirements inputs: User interview

Req. ID	Requirement
C.001	The tool will be run on individual feeds during the development process.
C.002	The tool will be run on entire production repositories.
C.003	The data itself (rather than the validation logic applied to it) must be reusable for other applications.
C.004	Verification must include both departmental- and project-level standards.
C.005	The tool must accommodate both XML- and database-output feeds.
C.006	Any errors detected must be captured and reported both for operational purposes as well as ongoing tactical process reviews and control.
C.007	The system must be able to run in the standard Data Engineering development environments, both on- and offshore.
C.008	The system must be able to run in the current production release of the platform.
C.009	The system must be able to run with minimal human interaction.
C.010	The system must be eventually integrated into the development suite.

Customer Requirements (Abridged)

Requirements: Functional Requirements

- Requirements inputs: Customer requirements, best practices documentation, project-specific implementations, user interview

Req. ID	Requirement
F.001	The system must produce a single summary file from multiple source materials.
F.001-A	The summary file must contain all data required for validation checks.
F.001-B	The summary XML document must conform to a standard format.
F.001-C	The system must be capable of reading from hierarchally-organized XML files in the file system.
F.002	The validation system must be configurable.
F.002-A	The configuration file must contain all data required for system configuration.
F.002-B	The configuration file must conform to a standard format.
F.002-C	The configuration file must be human-readable.
F.003	The system must be able to detect and report invalid Summary Values.
F.003-A	The system must be able to detect and report if a Summary Value does not exactly match a string in an enumerated list.
F.003-A-1	The enumerated list of values must be configurable.
F.003-A-2	The system must be able to select the appropriate enumerated list based on other criteria.
F.003-B	The system must be able to detect and report if a Summary Value does not match a regex pattern
F003.B-1	The pattern must be configurable.
F003.B-2	The system must be able to select the appropriate regex pattern based on other criteria.
F.003-C	The system must be able to detect and report if there is an incorrect # of occurrences of a Summary Value.
F.003-D	Negative Requirement: There is no requirement that the system use separate mechanisms for each of the Invalid Summary Value checks.
F.004	The system must be able to generate complex rules and apply them to Summary Values appropriately.
F.004-A	The system must accept configurable sets of rules in which the comparisons, operators, and values are customizable.
F.004-B	The system must be able to perform sequential comparisons on same or disparate Summary Values in order to apply rules.
F.005	The system must be able to generate reports with results.
F.005-A	The reports must include a "pass"/"fail" indicator for a given complex rule.
F.005-B	The reports must include the failed characteristic and failed value of any "fail"-status rule.

Functional Requirements (Abridged)

Requirements: Performance Requirements, Design Constraints

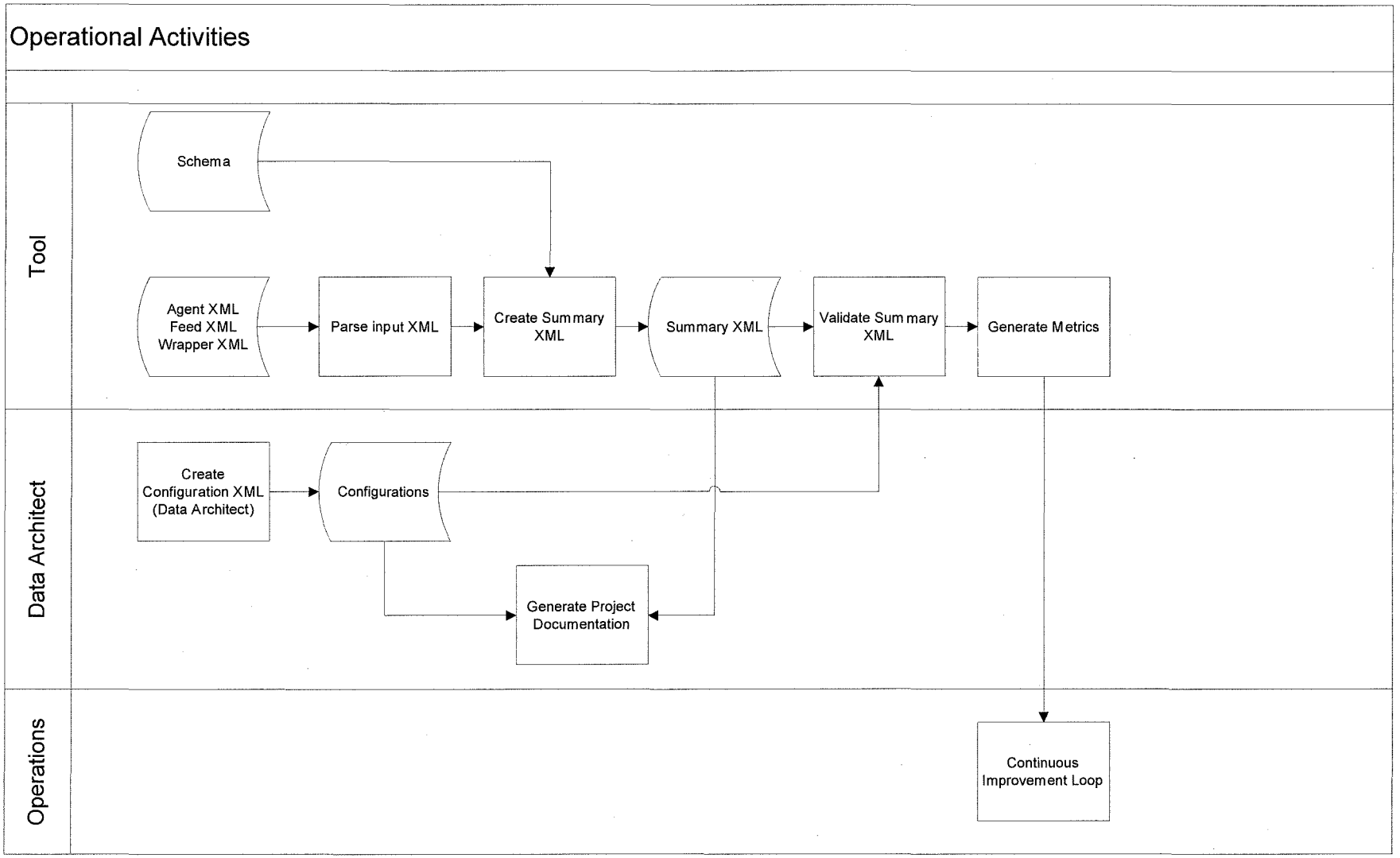
Req. ID	Requirement
P.001	The time to verify structural quality using the tool must be less than the manual equivalent.
P.002	The system must be able to operate across full production deployments consisting of up to 250,000 feeds.
P.003	The system must be able to perform at least 150 separate tests on each feed.

Performance Requirements (Abridged)

Design Constraints (Abridged)

Req. ID	Requirement
D.001	The system must adhere to all applicable coding standards (company, department).
D.002	The system must be architected in such a way as to be compatible with or complimentary to existing Data Engineering products.
D.003	The system must be written in Java.
D.004	The system must employ existing Java libraries when feasible.

Architecture



Technical Trade-Offs

- No general selection priority applied
  - Each functional need was evaluated in terms of the implementation trade-offs specific to that need.

Area	Advantages	Disadvantages
Off-The-Shelf Product	• No development cost • Low relative resource requirement • Product support	• Inflexible • Creates external dependencies • Workarounds, tweaking • Licensing, support costs • Unique codebase
New In-House Development	• Internal dependencies only • Flexible, agile, responsive • Correct solution • Easy platform integration • Library compatibility	• High relative development cost • High relative resource requirement • Internal support
Existing In-House Development (Data Engineering)	• Low development cost • Quick development • Smooth integration, support	• Strong dependencies with other applications
Existing In-House Development (Software Engineering)	• Low development cost • Strong platform integration	• Long lead time

Technical Trade-Offs

Agile Software Development

- Adaptive (rather than predictive)
- Emphasis on
  - Quick developer cycles
  - A constant working software suite, frequent deliveries
  - Responsiveness to change
  - Self-organizing teams

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
- The Agile Manifesto

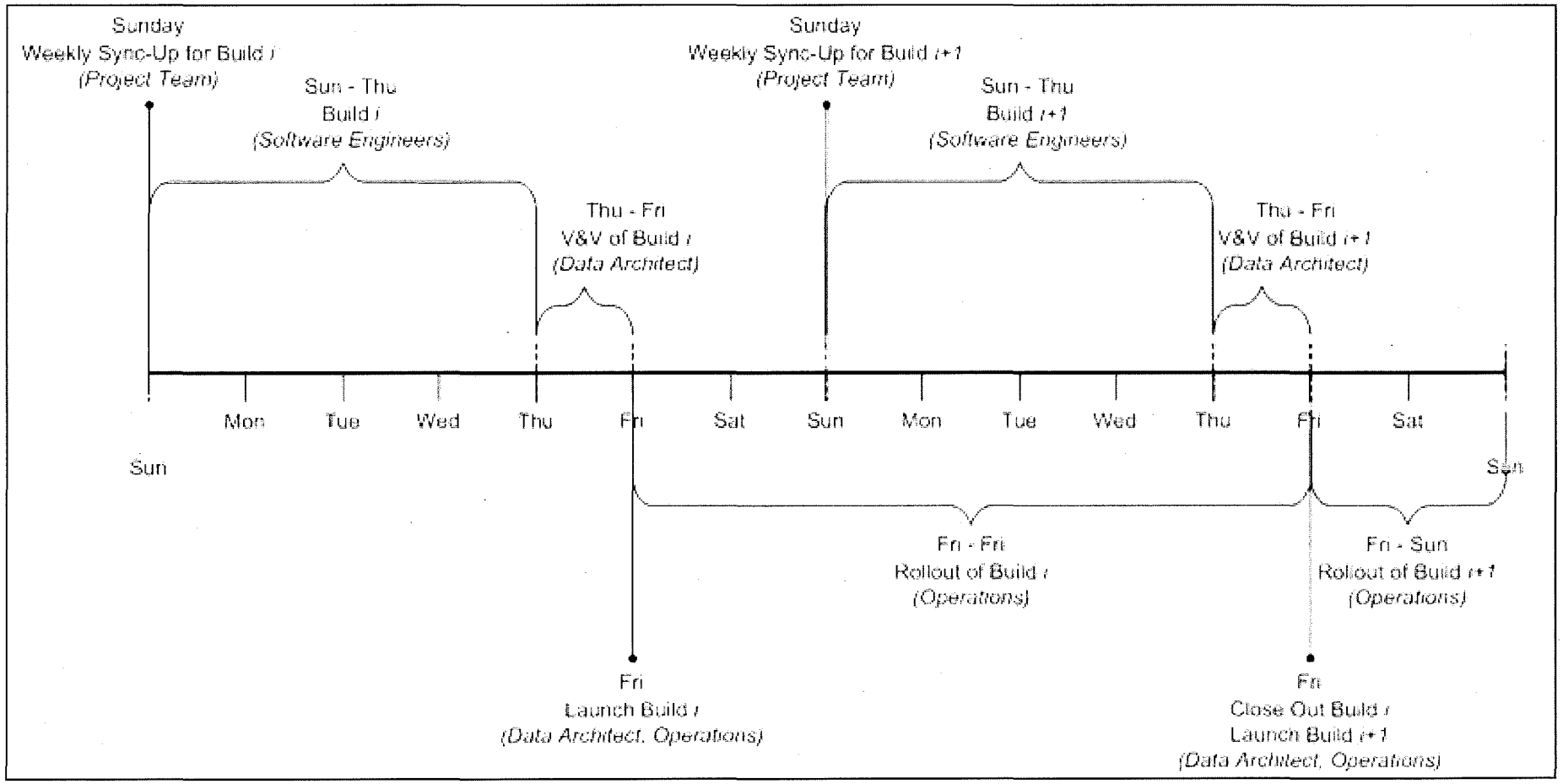
Area	Adaptive ("Agile")	Predictive ("Planed")
Criticality	Low	High
Requirements	Volatile	Relatively stable
Developer Experience	Senior	Junior
# Developers	Small	Large
Culture	Thrives on chaos	Demands order

- "Cowboy Coding"

Implementation

- Individuals and Interactions
  - Weekly Sync-Up
  - Daily "Stand Up" (10 minutes): What I did yesterday; What I am doing today; What, if anything, is blocking me
- Working Software
  - Weekly deliveries of fully-functional standalone incorporating all latest additions
- Customer Collaboration
  - Immediate deployment to production test beds
  - Fold-in of modifications, new requirements
- Responsiveness to Change
  - Completion of feedback loop
  - Real-time flexibility with regard to resource allocations and competing initiatives

Development Cycles



Data Collection and Evaluation

The Results

Data Collection

- Data Collection Methodology
  - Target production pipeline consisting of 45 feeds
  - Structural validation process includes 25 validation checks
  - Side-by-side manual and automated validations performed
- Limitations of Data
  - Short time period
  - Limited deployment
  - Incorporation of strategically-mandated operational workarounds
  - Inappropriately granular level of detail requested

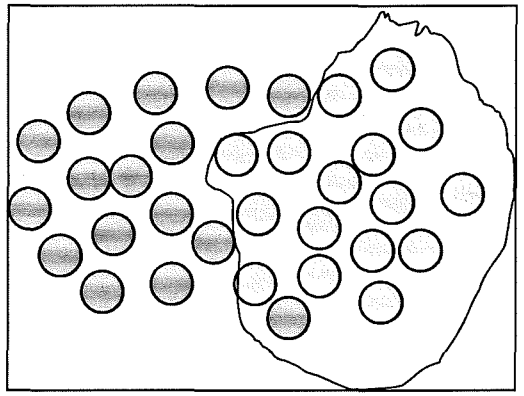
Findings: Effort

- Reduced to 3% of original person-effort
  - May be more: "Check" time is hands-on for manual process; the "Check" time for the tool is non hands-on.

Area	MANUAL Time (s)	WITH TOOL Time (s)	SAVINGS
Set-Up	30	120	
Checks	13500 (300 per feed)	245	
TOTAL	**13530 seconds** **3.76 person-hours**	**365 seconds** **0.1 person-hours**	**13165 seconds** **3.67 person-hours**
Assumptions: Pay Rate: $25/hour (on-shore resourcing)			
COST (Labor)	**$93.96**	**$9.53**	**$91.42**

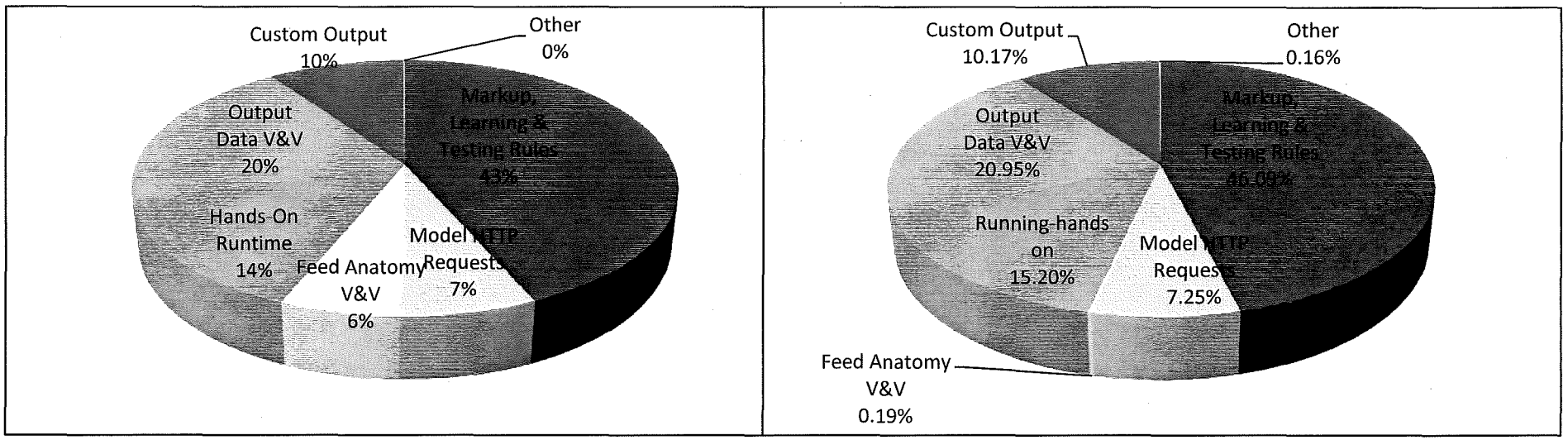
Findings: Precision and Recall

- Increased Accuracy Over Manual Process
  - Six structural flaws *undetected* by manual process
  - Successfully captured by tool
- Limitation: Alerting on non-alert conditions
- Precision and Recall
  - Recall: Number of relevant entities retrieved per number of relevant entities overall
    - Must be 1.0 for the tool to be *successful*
    - All error conditions detected
  - Precision: Number of relevant entities retrieved per total number of entities retrieved
    - Must approach 1.0 for the tool to be *efficient*
    - As few false positives as possible



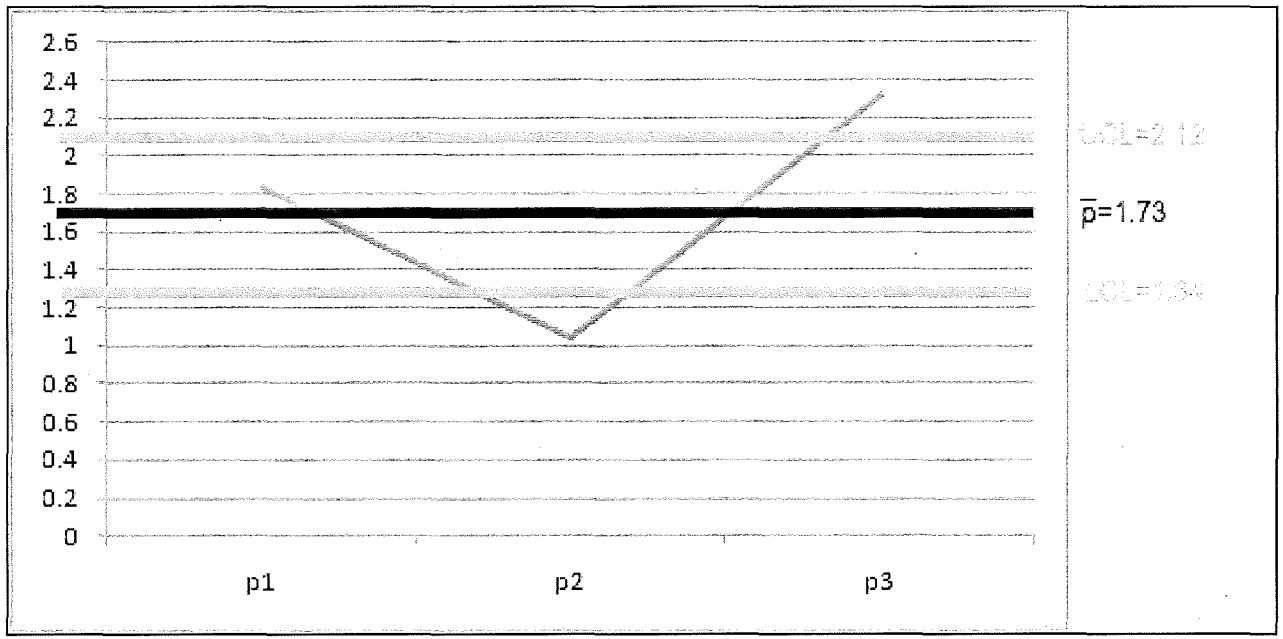
Findings: Efficiency and Scalability

- Scalability
  - Sample pipeline consisted of 45 feeds only
  - Average pipelines contain hundreds
  - Performance of automated solutions scales!
- Efficiency
  - Multiple points in the agent development process require verification of structural quality
  - Savings scale throughout process



Statistical Process Control

- Goal: Implement a Statistical Process Control for the agent development process as part of January 31, 2010 launch
- Feed Anatomy Tool provides infrastructure for data collection and reporting to support SPC



p-Chart based on initial data

Current Status, Next Steps, Conclusions

Conclusions

Current Status

- Feed Anatomy Project: Preliminarily successful!
- Already in production
- Already saving person-hours
- Closeout on target for January 31, 2009

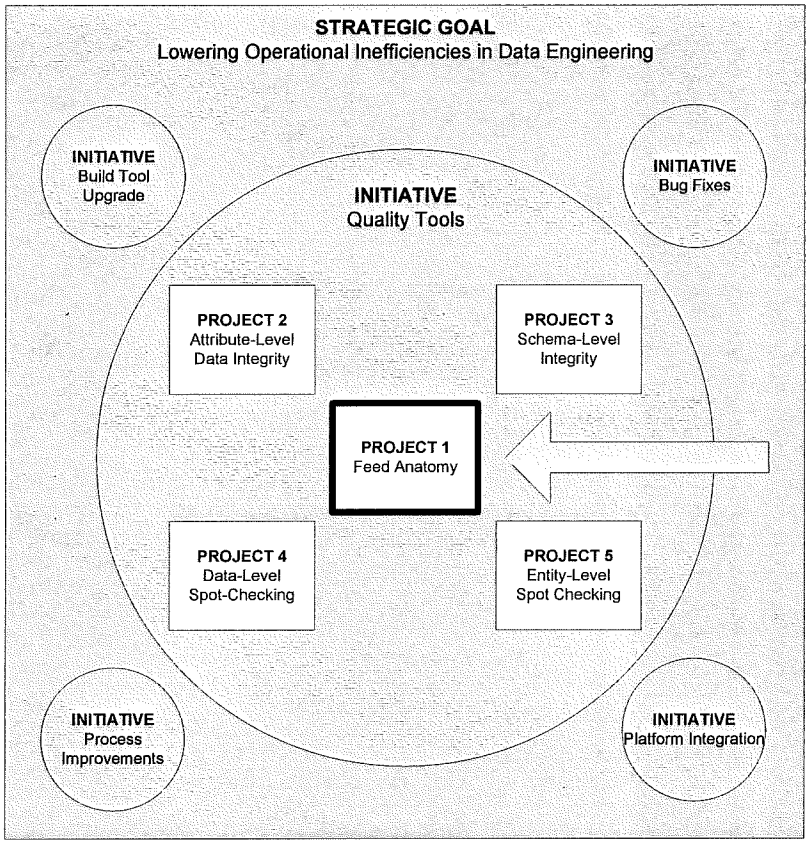
Next Steps

- Near Term: Feed Anatomy
  - Process integration: Deployment to full-scale production use across deployments
  - Process control: Implementation of monitoring/reporting feature to allow for statistical process control
  - Continuous improvement:
  - Platform integration: Integration into the platform architecture
- Medium Term: Attribute-Level Data Integrity, Schema-Level Data Integrity
- Longer Term
  - Entity-Level Spot-Checking
  - Integrated Web-Based Spot-Checking
  - Standardized quality methodology and certification

... but this is only the beginning!

Conclusions

- Powerful Combination: Systems Engineering Concepts and Strategic Alignment
  - Company-wide alignment
  - Powerful tools that produce results that are meaningful and *visible* across all levels



Thank You.