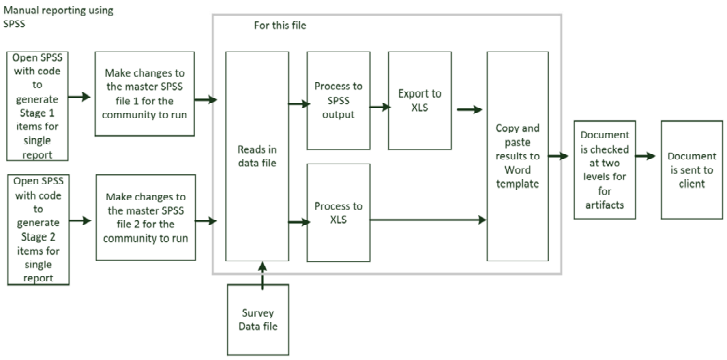
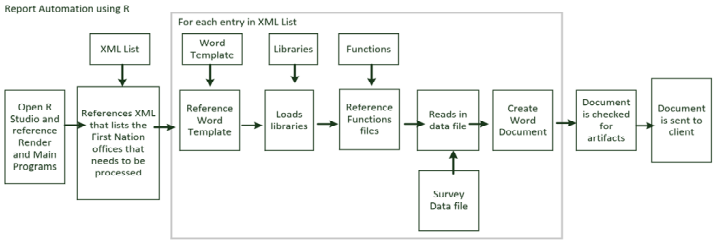




Digital Report Workflow Optimization

Problem: Manual approvals caused delays, manual editing takes time

Action: Mapped value stream, reduced bottlenecks, automated workflow

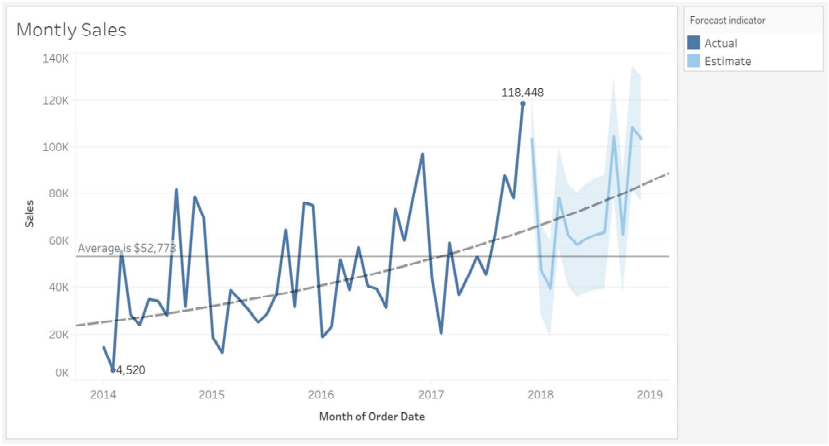
Result: Cycle time reduced 7 days → from 50 hrs to 20 minutes per report; 70% fewer steps; 96% productivity improvement

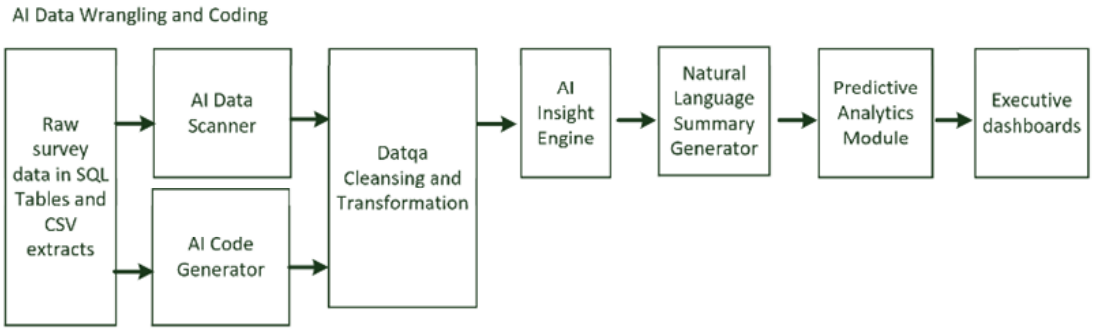
Up to Minute Analytics-Driven Sales Reporting

Problem: Information in silos and reporting had to be pieced together

Action: Designed monitoring dashboards for sales performance reporting with automated alerts, KPI's, and predictive metrics on desparate data

Result: On-time delivery +24%; proactive decisions





AI Programming and Data Wrangling

Problem: Minimal data cleansing; slow manual programming

Action: Prototype preliminary base code. Technology used: SQL Server, Python (pandas, scikit-learn), Large Language Models (LLM-based insight engine), Power BI for dashboards

Result: First automation realized a week or more savings of coding time and basic analysis of data

Portfolio - Allan Collier

Project	Smart Weather Portal
Stake holders/Customers/s	BigBucks Cafe
Created By	Fred Manager Product Owner
In JIRA	

Epic		Ver.	Summary - User Story ID	Summary Feature/Title	Description User Story (Value Statement)	Story points estim	Sprint	Reason	Developer subtasks	Definition of Done (User Story)	Definition of Done for Sprint	Definition of Done (Release)
For a weather portal to continue to provide an exceptional experience and satisfy the version launch needs, the highest Business Value will be when personnel from the BigBucks café is able to manage day to day operations and customize their products according to the predicted weather from the portal. **Software_epic_001**	1	1.01	US001	New user can register and log in on the portal	As a First time visitor to the Smart Weather portal, I want to be able to register on the portal, So that I can be notified of the weather conditions to plan day to day operations and customize product.	4	1	Has to be completed first to get access to the system	1. Write software 2. Create Test Cases 3. Review Test Cases 4. Execute Test Cases	1. User Story is fully implemented and requirements met (Acceptance Criteria) 2. Code is completed as described and will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server	1. US001, US002, US003 and US004 User Story requirements are met (Acceptance Criteria) 2. Code will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server	Version 1.01 – BigBucks IP to priority BigBucks Café requirements - This version is complete when all of the 7 User Stories have satisfied their individual Definitions of Done, the software has been QA tested as a whole against test cases, customer has approved the software demo, and localizations (language translations) has been completed.
	2	1.01	US002	User can be notified of the degree of rainy weather for the next week.	As the Coffee Shop Manager, I want to be able to determine rainy days for the week, So I can make available sealing stir sticks to ensure rain does not get inside drinks.	3	1	Had to be prioritized toward the top as it is most prevalent weather we face.	1. Write software 2. Create Test Cases 3. Review Test Cases 4. Execute Test Cases	1. User Story is fully implemented and requirements met (Acceptance Criteria) 2. Code is completed as described and will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server		
	3	1.01	US003	User can be notified of wind speeds for the week.	As the Coffee Shop Manager, I want to be able to determine the wind speed conditions, So I can determine if I need to make available aggressive lids for the cups.	2	1	Had to be prioritized toward the top as it is most prevalent weather we face.	1. Write software 2. Create Test Cases 3. Review Test Cases 4. Execute Test Cases	1. User Story is fully implemented and requirements met (Acceptance Criteria) 2. Code is completed as described and will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server		
	4	1.01	US004	User can be notified of clear weather for the next week.	As the Coffee Shop Manager, I want to be able to determine clear days for the week, So I can plan for outside seating at the sidewalk café bar.	1	1	Had to be prioritized toward the top as it is most prevalent weather we face.	1. Write software 2. Create Test Cases 3. Review Test Cases 4. Execute Test Cases	1. User Story is fully implemented and requirements met (Acceptance Criteria) 2. Code is completed as described and will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server		
	5	1.01	US005	User can be notified of temperatures for the next week.	As the Coffee Shop Manager, I want to be able to determine hot days for the week, So I can source necessary iced beverage products to sell.	2	2	There tends to be less prevalent days here.	1. Write software 2. Create Test Cases 3. Review Test Cases 4. Execute Test Cases	1. User Story is fully implemented and requirements met (Acceptance Criteria) 2. Code is completed as described and will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server	1. US005, US006, and US007 User Story requirements are met (Acceptance Criteria) 2. Code will build without warnings 3. Code unit tested 4. Documentation updated 5. Build Pushed to demo server	

Epics created for Agile Sprints for Weather Applicaton Development

Problem: Satisfying user needs for Weather Application to help with event planning

Action: Created several Sprints to develop the weather app software

Result: Allow for better weather predictability to plan for outdoor events

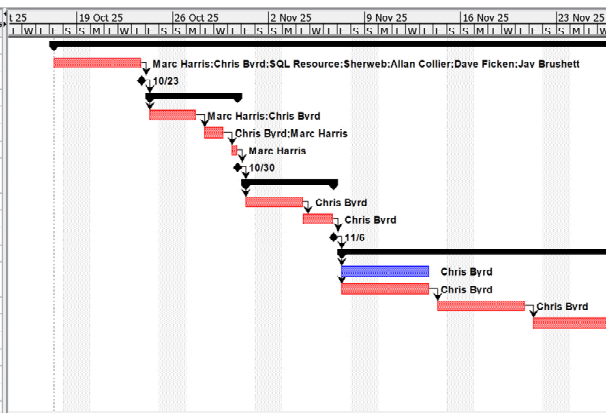
Waterfall

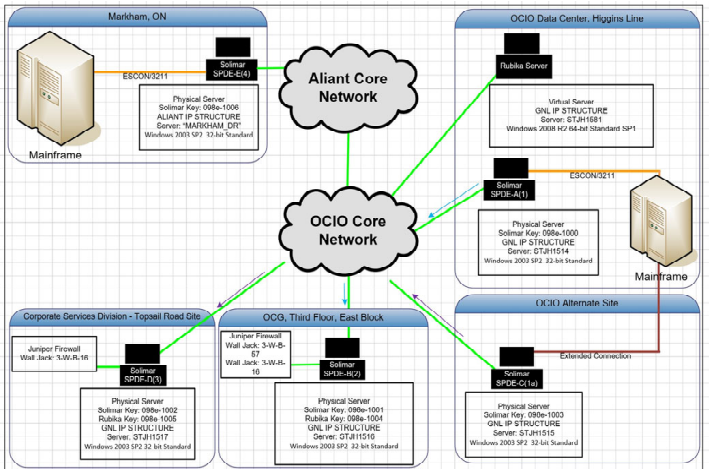
Problem: Vendor had no structure for RFP, Project Plan, or Proposals for On-prem network to the Azure Cloud

Action: Created high level timeline, however made more granular for implementation pieces

Result: On-time delivery, zero implementation delay, and facilitated Upper-management reporting

	Name	Duration	Start	Finish	Predecessors
1	**AZURE Migration Project**	35 days	**10/17/25, 8:00 AM**	**12/4/25, 5:00 PM**	
2	Project Discussion	5 days	10/17/25, 8:00 AM	10/23/25, 5:00 PM	
3	Project Acceptance	0 days	10/23/25, 5:00 PM	10/23/25, 5:00 PM	2
4	**Phase I: Discovery and Design**	5 days	**10/24/25, 8:00 AM**	**10/30/25, 5:00 PM**	**3**
5	Azure tenant setup	2 days	10/24/25, 8:00 AM	10/27/25, 5:00 PM	3
6	Network Design	2 days	10/28/25, 8:00 AM	10/29/25, 5:00 PM	5
7	Cost Finalization	1 day	10/30/25, 8:00 AM	10/30/25, 5:00 PM	6
8	Discovery and Design Completed	0 days	10/30/25, 5:00 PM	10/30/25, 5:00 PM	7
9	**Phase II: Test VM Pre Deployment and Testing**	5 days	**10/31/25, 8:00 AM**	**11/6/25, 5:00 PM**	**8**
10	Validate connectivity	2.5 days	10/31/25, 8:00 AM	11/4/25, 1:00 PM	8
11	Test firewall rules	2.5 days	11/4/25, 1:00 PM	11/6/25, 5:00 PM	10
12	Predeployment and testing completed	0 days	11/6/25, 5:00 PM	11/6/25, 5:00 PM	11
13	**Phase III: Migration and Execution**	15 days	**11/7/25, 8:00 AM**	**11/27/25, 5:00 PM**	**12**
14	SQL Server Migration	5 days	11/7/25, 8:00 AM	11/13/25, 5:00 PM	12
15	Web Server Migration	5 days	11/7/25, 8:00 AM	11/13/25, 5:00 PM	12
16	vMX deployment	5 days	11/14/25, 8:00 AM	11/20/25, 5:00 PM	15
17	FortiGate deployment	5 days	11/21/25, 8:00 AM	11/27/25, 5:00 PM	16
18	**Phase IV: Validation, Testing and Tuning**	5 days	**11/28/25, 8:00 AM**	**12/4/25, 5:00 PM**	**17**
19	Performance testing	2.5 days	11/28/25, 8:00 AM	12/2/25, 1:00 PM	17
20	Documentation	2.5 days	12/2/25, 1:00 PM	12/4/25, 5:00 PM	19
21	Project Complete	0 days	12/4/25, 5:00 PM	12/4/25, 5:00 PM	20





Network Architecture

Problem: Solution needed for communications when mainframe was moved from Newfoundland to Montreal

Action: Strategized and architected network communications to allow Newfoundland access to remote mainframe from network.

Result: Remote access, less cost to operate and more efficient operations