



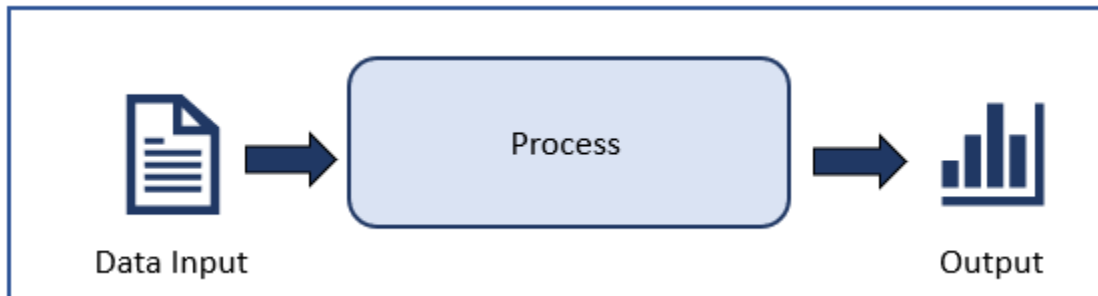
Mississippi Department of Education

Automate: A Playbook for SEA Data Processes

Deborah Donovan
ddonovan@mdek12.org
August 2021

Introduction

In State Education Agencies (SEAs), the availability and accessibility of high-quality data is imperative for effective decision-making, evaluation, and accountability. High-quality data means data that is timely, accurate, complete, well-defined, and well-understood. In an SEA, much of this data is collected from school districts. Whether once a year or once a minute, data transferred from a district to an SEA is vital to the success of producing outcomes based on high-quality data. Many of these data are used in data processes. A data process is any form of data ingestion, mechanisms performed on that data, and outputs produced by those mechanisms.



When one or more parts of a data process includes manual interventions, a siloed data process is created. The providers of the data (usually districts) are minimally involved and have limited or no visibility into the process.

Examples of manual interventions

- Manipulation of data sources and inputs – this can include data cleansing, filtering, or the creation of data sets based on business rules that are not aligned to the current goal
- Calculations – the most common practice observed is data that is exported, manipulated (calculated, aggregated, etc.), and returned to a system or a user
- Displays – data displays such as reports or visualizations that apply filters or calculations, essentially creating new result sets

Automating a data process means that the process is defined, repeatable, and trustworthy. In other words, you can expect consistent results each time without manual interventions. While it is never the intent to purposefully create siloed processes, there are several factors that may lead to this inevitable outcome. These factors are often capacity constraints of staff, time, and technology. Automating and integrating these siloed processes requires the availability of the same capacities. By using these capacities wisely, automation of data processes can lead to increased efficiency, transparency, and trust between an SEA and its stakeholders.

This playbook provides a framework for automation of data processes that addresses challenges frequently faced in the SEA environment. These challenges are likely to exist in local education agencies and other public sector environments and similar approaches would prove beneficial.

A Case Study in Automation

At the Mississippi Department of Education, a large amount of data is stored in a single repository, the Mississippi Student Information System (MSIS). And for the most part, this system is used to generate

reports and calculations. But over the years, silos of data and processes have been created in the agency. These have been largely in the areas of accountability and assessment data sets, as these departments typically have data and research resources as part of their teams.

Contributors to siloed processes:

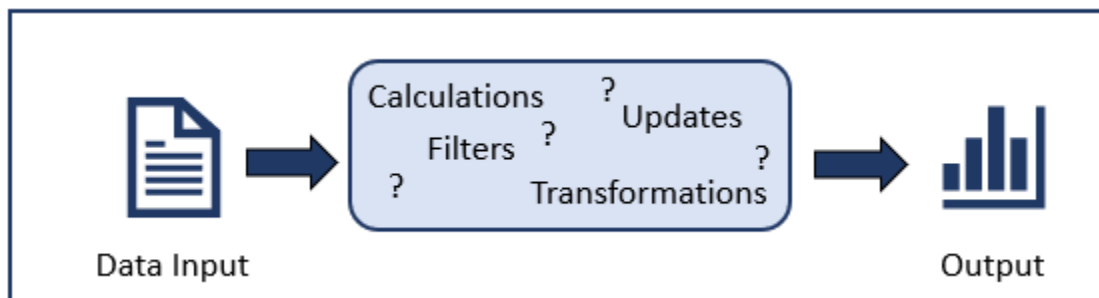
- Complexity – Complexity of calculations or complexity of business rules lead to processes that are best understood and implemented within a particular program area.
- Urgency – Short timelines can lead to ad-hoc processes created to get past the deadline and meet the requirement, often with the intent to re-visit and further develop these processes. Alas, there always seem to be higher priorities and the next urgent project that leave the process in its ad-hoc state year after year.
- Staff - The lack of needed staff or staff availability in the required time can lead to data “heroes” stepping in to build the required process in isolation. These heroes typically have subject matter expertise and can work independently to build a solution. This also leads to processes that are highly dependent on specific individuals.
- Systems – At the time processes are created, there may be a lack of system capabilities that would allow the process to be integrated into larger systems.

Processes that are built under these constraints often lack transparency, especially for school districts. Districts submit data to the SEA and months later, a report emerges that has little relation to the data originally submitted. These are referred to as “black box” operations – data goes in and results come out, with little understanding of what happens in between. Districts find it difficult or impossible to produce the same outcomes using the data they provided.

Be Considerate of Data Heroes

Data heroes step in and “get the job done” when the stakes are high. Getting the job done rarely means that the process is documented or perhaps even shared with others. Gaining the reputation of a data hero can also lead to frustration, burn-out, or worse when they are too-often asked to step in and save the day. It is not a sustainable model for work and leaves the agency at higher risk of orphaned data processes when the hero inevitably decides to hang up their cape.

A “Black Box” operation



The 4-year cohort graduation rate calculations were one of these processes in Mississippi. The graduation rate is based on a 4-year cohort of students and takes all student enrollments, transfers, and exit statuses into account when calculating the rate. It is also a complex calculation because of its longitudinal nature. Students are assigned to the school where they were enrolled the longest, which may not be the school they graduated from.

The agency had little documentation on the process and districts were having a hard time validating or replicating their results.

This is also a high-stakes data point in Mississippi and is part of each school and districts' accountability rating. Districts often run their own projections and estimates of graduation rates to monitor their performance. When results were being produced from the SEA, they were not matching district's local calculations. Each party believed their results were correct, but neither could be truly validated because the business rules were not detailed and there was no way to determine exactly how each student was contributing to the rate. Districts received a student level data file, but without context or explanation of the statuses.

The lack of transparency and trust led to a call for the SEA to investigate the current calculations. The SEA worked with selected districts and investigated individual cases. When the process was manually followed using the data and the calculations, it could be determined how the eventual status for a student was determined. There was a great deal of confusion surrounding the business rules. For example, understanding how students who transferred to GED programs were categorized. Because the SEA was able to manually trace the data through the calculations, there was no reason this could not be automated for every student.

The SEA mocked up reports that displayed intermediate calculations and categories and shared them with the districts to solicit feedback. Thus, the journey to automate the calculation began.

In the current process, data was being extracted from the student system, MSIS, imported into a statistical package, calculated, and reported out in summary reports. Since MSIS contained all the necessary data inputs, the calculation was able to be automated within the system, which provides better reporting and transparency opportunities.

As a result of the automation and improved reports, districts had confidence in the results, trust was restored, and a greater sense of collaboration was beginning to build around data processes.

Benefits of Automation



Transparency

Stakeholders have full access to processes performed with their entrusted data



Trust

Increased trust between agency and stakeholders built on transparency and stakeholder engagement



Capacity

Time and resources saved from manual processes as well as time saved investigating issues and tracking down errors/bugs



Accountability

Increased accountability and confidence as guardians of the data and the process



Growth

Ability to capitalize on increased capacity, trust, and goodwill by offering high-value services

The steps below are a guide to selecting and automating data processes. Mechanical steps such as implementing technical solutions and creating project plans are provided as examples of a typical software development lifecycle. These steps should be adapted to best fit your SEA's standards and preferences.

Step 1: Define the Problem

Every agency has constraints on their capacity to improve existing systems. This includes everything from resource and time constraints to competing priorities. While a detailed project portfolio with priorities and linked resource availabilities is the ideal solution for planning projects, it has a high overhead costs that most agencies are not able to consistently maintain. A more manageable solution is to categorize work and plan smaller groups of initiatives. For example, you may be able to systematically plan and prioritize Federal reports and system enhancements based on annual calendar loads. Enhancement projects such as data automation projects can be planned for the off season of high stakes reporting.

Beyond general capacity and timing constraints, the next hurdle to approach is identifying potential candidates for automation.

Considerations for automation

- System or component?
 - Does an entire system, including collection and reporting, need to be automated?
 - Would automating a particular component, such as automating a manual data entry process, deliver similar benefits?
- Inward or outward facing?
 - Does the process have an outward facing component, such as district or school level reporting?

- Automating internal processes will increase capacity but may not impact transparency or trust.
- High stakes?
 - Automating a high stakes process can mean assuming a high risk, but the risk of a poorly implemented high stakes process is often higher.
 - Is the process closely aligned to the SEA's strategic plan?
- What do you need to automate?
 - Is the process within your control or purview?
 - Do you have authority and access to the data you need?
- What are the concerns with the current process?
 - Is the current process untimely, unclear, or error-prone?
 - The more concerns, the higher priority
- Dependent on individuals?
 - Is the current process sensitive to personnel changes?
 - Do the personnel involved have competing priorities?
- Related impacts?
 - Would a delay or failure of the current process impact other projects or other initiatives?
- Related opportunities?
 - Are there identified changes such as new requirements or changes to business rules to the current process?
 - Will you be modifying the process anyway?

Step 2: Current State

Once a process has been identified for automation, it is tempting to begin planning the technical solution immediately. However, resisting this temptation can prevent you from turning a bad manual process into a bad automated process. Understanding and documenting the current process will inform the best possible future solution.

The first step is to identify the owner of the process. The success of the automation project will depend on the owner's willingness and ability to play a critical role and engage with the automation team throughout the project.

The next step is to identify what currently works well within the process and should be preserved. Similarly, you will need to identify the pain points with the current process. You have likely already gathered much of this information when the process was identified as a candidate for automation. This is also an opportunity to create an

No Documentation? No Problem.

With the owner of the process, create a process map. A process map is a visual representation of the roles, inputs, activities, and outcomes of a process. By asking the owner simple questions such as "How do you start?" and "How do you know when you're done?", you can begin filling in the gaps of what happens in between. By involving staff who are involved in the current manual process, you can piece together all components of the process.

What if the current owner left?
We've all been there. You've been asked to fix that process that's been auto"magically" running on the mainframe for years. Or the code without any comments. Or the set of instructions that don't make any sense.

Certainly, this is a bigger challenge, but not all is lost. Start with the existing code and start translating to pseudo-code. Keep translating and documenting until you reach a business-level of process steps. Then, start researching those steps with the process owner to understand the code. It's tempting to abandon the old code and start fresh, but there are often undocumented and un-remembered business rules to be discovered.

honest dialogue with the stakeholders involved in the process. It is important to involve them at every step of the project. Transparency within the automation is just as important as transparency of the process that is being automated.

The final step is to gather all existing documentation for the process and build a complete picture of the current state. Engaging with the stakeholders around the current process will likely have revealed new requirements that should be captured and curated in the next steps.

Step 3: Define the Future State

Before defining the “what” to be done, you must understand the “why”. Defining the use case for automation identifies the problem(s) that are being solved by automating. Although automation on its own can build trust in a process in that it produces consistent and repeatable actions, it does not address the less tangible challenges of perception and culture change. Think of this step as a goal-setting activity. What will it look like when you succeed (What is your definition of success)? And how will we get there?

Now it is time to start documenting plans and the proposed solution. Start from the highest level of detail to the most detailed.

- What are we doing? (Functional Specifications)
 - What are you trying to accomplish?
 - Use language that infers automation. Instead of writing “Office of Data will create report of teaching vacancies.”, you would say “Office of Teaching will access teacher vacancy report to determine shortage areas.”
 - Reference each business rule to ensure they are addressed in the solution.
- How will we do it? (Technical Specifications)
 - Documents the technical steps needed to put the plan into action
 - Use the current state documentation or solution for reference, but not guidance. Make room for innovation and improvement. The development team should be given latitude to design the best solution, so make sure they are the primary authors of this document.
- Who will do it? When will they do it? (Project Plan)
 - Using the technical specifications as a guide, plan out the work taking into consideration capacities, availability, and feedback cycles needed.
 - Clearly define and communicate the roles and responsibilities of everyone involved

Validation Checkpoint
<ul style="list-style-type: none">▪ Look back at your use cases to make sure the solution still reflects the business case▪ Compare the current state documentation to future state design to identify gaps▪ Validate solution with stakeholders using mock-ups and visual aids

Waterfall or Agile?

One of the most popular and oldest software development lifecycle approaches is known as “waterfall”; an approach where requirements are gathered and fully defined before software is written. The more modern approach of Agile methodology uses an iterative approach of working with stakeholders to deliver portions of the solution in smaller sprints. This allows the stakeholder to have continuous feedback throughout the process.

This document describes a blended approach of waterfall and Agile. Since the process being automated already exists, the gathering of business rules and other documentation is largely completed at the beginning of the process (waterfall). The stakeholders in the process remain continuously engaged in all stages to validate and confirm the intended outcomes and to provide feedback on the value of the project (Agile). The steps may be adapted to fit your agency’s established approach.

During this stage, it is important to involve the owner of the process, subject matter experts, and stakeholders. They will be able to validate the solution, correct any errors in translation of business rules, and even point out mistakes or errors in logic inherited from the original solution. There are likely to be changes or enhancements needed to the old solution, so you should take advantage of every touchpoint with the stakeholders to identify and incorporate those changes into the new solution.

It is also important to evaluate and confirm the commitment between the technical and business staff to the shared vision. With all change, the feelings of fear and uncertainty must be balanced with both perceived and actual benefits. Lingering doubts or distrust can manifest as conflict later in the process, leading to delays, finger-pointing, or worse.

Step 4: Execution

Now the coding can begin. However, this doesn’t mean that the development work is handed off to the development team and is delivered back to the owner and stakeholder when completed. Stop and validate the work in progress at pre-defined intervals. These intervals may be based on time, milestones, drafts, mock-ups, or other deliverables. This iterative process creates a continuous feedback loop that improves the final product and more importantly, fosters trust and transparency in the process.

It is extremely important that all members of the development team understand the importance of these validation checkpoints. On one hand, freedom and trust should be given to the team to innovate and define the best solution to meet the needs. On the other hand, assumptions made about the preferences and needs of the stakeholder or owner can become barriers to a successful product.

One approach, borrowed from Agile methodology, is the idea of including a “product owner” on the team. The product owner is a person that understands and represents the interests of the stakeholder and can advise the development team to make the most appropriate decisions.

It will be necessary to decide when internal vs. external stakeholders should be involved in reviews and input sessions. There will be certain deliverables that make more sense, such as reports and visualizations, to bring to external stakeholders. Complex legal and statutory requirements may be best dealt with by internal stakeholders.

In any case, the development team should never work in a silo.

Step 5: Test, Test, Test

In addition to the intermediate validation that took place throughout development, testing should be incorporated into each possible iteration of the development process.

Types of testing

- Unit testing – checks that a single function or component works as expected; usually performed by developers
- Integration testing – checks that multiple functions or components work together as expected; usually performed by the development team
- Functional testing – checks that the solution meets expectations of the design and produces the correct output
 - Use multiple years of data, when possible, to capture edge cases
 - Usually performed by a combination of development team and internal stakeholders
- Acceptance testing – checks that the end user is satisfied and accepts the solution

Another type of testing specific to automation projects is the replication of the manual or existing process. Ideally, the results should be the same. If not, it is critical to identify the cause for the differences and ensure that the new solution is applying the business rules correctly and can be validated by the owner and stakeholders.

Do not hesitate to share early drafts, incomplete, or unpolished with stakeholders to make sure you and the development team are on the right track.

Step 6: Roll out

Once you are confident in the results and all testing has passed, it is time to plan the roll out of the final process.

Stakeholders such as schools and districts are especially useful to involve in this process. Many will be eager to volunteer to gain early access to reports or results and give feedback on behalf of their peers. Don't be afraid to engage with your harshest critics in this space. Those who are most vocal about the problems with the current process can often be the most willing to engage in making the process better.

Planning for a successful roll-out includes training opportunities and awareness campaigns. Delivering data without context and training will counteract the resource savings you are attempting to achieve by automation.

Step 7: Reflect

Part of the continuous improvement process includes reflection. After the roll-out of the automated process, hold sessions with as many stakeholders as possible to reflect on lessons learned. This is common practice with software development teams, but the process should be extended beyond those directly involved in development.

Lessons learned identify:

- What went well (Repeat these things)
- What could have gone better (Revise these things)
- What will be done differently next time

This is also the time to revisit documentation. It is likely that something in the current or future state documentation changed since it was developed, so now is the time to update that documentation. At a minimum, the documentation should be updated with maintenance and deployment instructions. For example, what codes or data sets must be refreshed or reviewed and on what frequency once the automated solution is deployed.

Next Steps

You've spent a lot of time and effort building trust through this process. Use that trust to build upon your success and engage in the next project that was perhaps a bit too risky to take on before. The advantages of creating a culture of collaboration cannot be overstated. Every process that's automated creates an opportunity and space to bring more value to your stakeholders.

If this is an automated process, that means it will run more than once. And something will eventually need to be changed or updated. If you don't update your documentation, you are feeding into the same system that caused you to be here now.

Coding and development are very small parts of an automation process. The success of automation processes relies on understanding people and processes and how they intersect with technology. It's more about cultural standards than data standards. And it's more about recognizing and acknowledging the value in your co-workers and stakeholders that was once clouded by frustration.