Finance uses anomaly detection and automation to transform royalty statements processing

Like many companies, Microsoft pays royalties for games, videos, content, software, and other creative work used in its products and services. There is a very high bar for accuracy when processing royalty payments. Operating under strong controls and the compliance tone set by Microsoft Finance, the process requires reconciliation at multiple levels, cross-checking, reviews, and payment approvals. With more than 30,000 statements—worth nearly $5 billion—manual processes are time consuming and costly. The Finance Operations team partnered with Microsoft Core Services Engineering and Operations (CSEO) on a modernization initiative to help streamline the process through technology while further strengthening the Sarbanes-Oxley Act (SOX) and operational controls that are required for these types of workflows.

We worked with a core team of royalties-process subject matter experts from Finance Operations to develop a three-part solution that automated royalty statement reconciliation, utilized machine learning to detect anomalies, and automated the approvals process for more than 50 percent of the royalty statements being processed—saving thousands of hours each year.

Finance Operations collects all transactions that are subject to royalty payments based on views, downloads, or plays, depending on the type of content. Then, the payment amount is calculated according to the contract with the content owner, a statement is created, and the Finance Operations team ensures that the payments are processed. Due to the high dollar amounts and strong SOX control requirements, business users and financial analysts had to manually reconcile multiple data sources in order to provide a high level of confidence and assurance in the final calculated amount.

This was the first time we were considering deploying machine learning and automating important finance processes that are regulated by federal SOX rules. We had to partner closely with internal Controls and Compliance teams, along with the Corporate Finance Controls group, to ensure that risks were properly addressed. Our goal was to not only meet current SOX requirements, but to ultimately enhance them.

We focused on the three parts of the royalty workflow for which we thought automation and machine learning could provide the most time and effort savings:

• **Three-way reconciliation (RECON).** The reconciliation process was completely manual, time consuming, and involved multiple people in multiple regions relying on Microsoft Excel workbooks and well-documented processes. We used Microsoft Azure HDInsight to automate the process of comparing the various data sources that are used to reconcile or verify the information that is included in a royalty statement.

• **Royalty Anomaly Detection (RAD).** It is not humanly possible to analyze the full range of historical data required to identify anomalies for every scenario. Numbers can acceptably deviate from their general range yet still be in line with what is expected at a certain time of the year, in a specific region, or in relation to another related product’s release. We implemented RAD using Azure Machine Learning to analyze two full years of historical data as well as current data to quickly and accurately identify anomalies, or outliers, given all acceptable variances.

• **Royalty Efficiency and Automation Project (REAP).** Many hours were spent waiting for users to approve statements. REAP has introduced automated approvals for royalty statements that have passed RECON, have a green (passing) RAD, and meet specific business rules. For example, currently, auto-approvals are available only for statements that are less than a defined dollar amount—an amount that made more than 50 percent of statements eligible for auto-approval. We do have plans to increase that threshold amount in the future.

Transforming how royalty statements are processed

We built an experience that automates the royalty statement approval process using cloud-based distributed systems performing at scale. It also creates better visibility and transparency within the statement portal into how reconciliation was performed to help ensure accurate payouts. We are now using automation and machine learning to save time—time that can be better spent on other activities—while providing the necessary process oversight.
Solution overview

As illustrated in Figure 1, the new workflow uses RAD, RECON, and REAP—effectively creating a fully automated “happy path” for auto-approval of royalty statements processed today.

Figure 1. End-to-end royalty statement workflow

Here’s how the end-to-end statement workflow operates:

1. The royalty calculation engine creates a statement from its results.
2. RAD detects a new statement.
3. RAD compares statement information to the previous 24 months of historical data and then records its observations.
4. Daily three-way RECON compares the calculation engine source, origination source, and a third independent source.
5. REAP performs a check to determine whether the statement is eligible for auto-approval.
6. REAP uses the outcomes of steps 2 through 5, which includes RAD observations, reconciliation results, royalty-earned amounts, and the statement details, and applies the business rules. If the statement is eligible, it auto-approves.
   a. If auto-approved, REAP auto-approves the statement for payment processing.
   b. If not auto-approved or not eligible for auto-approval, the statement is sent for manual review and approval.

Developing a RAD proof of concept

Beyond theorizing about how machine learning could accurately detect anomalies in a royalty statement or proposing how automated reconciliation would produce comparable results to manual reconciliation, we needed to prove to Finance that it could be done and that the results could be trusted. We needed to create a proof of concept. Taking the high-level business requirements and assumptions about the desired outcomes, we began working on a prototype for RAD.

Moving data into the cloud

Machine learning requires a lot of data to learn, and the data we needed was stored both in the cloud and on-premises. For the purposes of our proof of concept, we needed to move all the required data into the cloud. To ensure that we had a big enough data sample for the machine learning, we loaded 24 months of historical payment and transaction data into an HDInsight cluster and we used business rules to filter the royalty-bearing transactions to ensure parity. The historical data included transactions, payment history, seasonality, and launch events, which allowed the team to create a machine learning model that would include typical variances. As new data became available, we included it in the machine learning’s data set.
Translating business knowledge into a machine learning model

After we ensured that the data required for machine learning was available, we began working with subject matter experts in Finance Operations. We needed to really understand why some data points can be outliers, whereas others might look like outliers yet are within a normal range when other factors are considered. To be able to apply machine learning to determine anomalies in payment variances, we needed to understand the thought process and business rules used by analysts to determine the validity of a payment. The only way to distill that knowledge was to work with the people who had a deep understanding of how the process works, what data is required, how it is structured, and how it flows through the system.

That knowledge was translated into high-level requirements that were converted into statistical formulas used to identify the outliers. We provide the model with a set of inputs, which are run through our formulas, and then an output is produced. Because we weren’t using a named or built-in machine learning model, we worked with a data scientist to create the interquartile regression formulas for our model. The Royalties team expects payments for a contract to fall between a minimum and maximum range—data points outside of that range are outliers. We used interquartile regression because it allows us to clearly define the variance based on cause. Not only can we detect an anomaly, but we can determine which dimension caused it.

We then were able to use Azure Machine Learning Studio to share the workspace, work with our model, and quickly preview the results as we made changes.

Refining our results

When we began running our model, we received more noise than focused observations. We expected that would be the initial result given that a lot of data was in the system and data cleansing had not yet occurred. We had also introduced a lot of constraints, thinking that we should capture more results. As we slowly began cleaning up the data and working with subject matter experts to validate the outputs, we started fine tuning our models and began to see better results.

For almost eight months, we continually refined our results and sought feedback from subject matter experts from Finance Operations. Their feedback was invaluable; they understood the conditions and context around why numbers can spike or dip in relation to other variables. After they validated that the prototype’s anomaly detection results were meeting the high-level requirements, we moved the RAD prototype into the production phase.

Moving RAD into production

During the first phases of moving RAD into production, the outputs were validated and iterated upon. We spent several more months defining scenarios and working with the business to capture more detailed business requirements as well as adjust our machine learning model to meet the desired outcomes.

For every new transaction, we ran RAD to identify whether the current numbers were in line with the previous 24 months of data for the same combination, or whether it was an outlier. RAD also considers seasonality and launch periods before marking anything as an outlier. As illustrated in Figure 2, observations are then categorized as Unexpected Anomaly (red), No Anomaly (green), or Expected Anomaly (yellow) due to seasonal variation or new launch.

![Figure 2. Anomaly detection color-coded status](image_url)

As we used RAD to perform anomaly detection, Finance Operations was manually reviewing our results and providing feedback about why it agreed or disagreed with the output. That human feedback helped us to continue to refine our machine learning models.
Finance uses anomaly detection and automation to transform the happy path for royalty statements

When an anomaly is detected, additional dimensions such as seasonality, launch periods, and geography are used to isolate the source of the anomaly. Figure 3 illustrates RAD results displayed in the statement portal, with detection status and details.

Figure 3. A snapshot view of RAD status and an Anomaly detail view available in the statement portal

We continue to feed current transactions into the two years of historical data that the machine learning model is using to train itself. RAD is always evolving to better detect anomalies and reduce the risk of processing inaccurate royalty payments.

Automating three-way reconciliation

The manual reconciliation process was transformed into an automated one by bringing various streams of data into an HDInsight cluster and applying business rules on top of it. The data streams include contract information, product information, and transactions from the Microsoft sales system. The comparison results are provided at the statement level. Whether a statement has been enabled, or is even eligible, for auto-approval using REAP, it is still fed into three-way reconciliation.

During three-way reconciliation, source numbers, destination numbers, and a third independent source are compared programmatically, and the results are published every time data from a new statement is ingested. Each statement is then tagged with a data-quality score, depending on the reconciliation results, and the results are published in the statement portal.

For a statement to be reconciled through automation, it must satisfy two conditions for a period. If either condition fails, the reconciliation will follow a manual path.

- The majority of the paid quantity being calculated must be reconcilable.
- The total sum of the paid quantity of products found in the royalty calculation engine must be within the five-percent variance threshold unless the reconcilable quantity difference is less than or equal to 100—a configurable amount, set by the business.

Automating approvals with Royalty Efficiency and Automation Project (REAP)

To reduce the level of effort required to manually validate and approve statements, we re-engineered and streamlined the processes to enable the deployment of intelligent and automated solutions that enable touchless processing. We developed an orchestrator using Azure Data Factory to perform auto-approvals for statements that fall within certain parameters and are eligible to take the “happy path” to approval. Today roughly 50 percent of all royalty statements are truly automated, from the point of statement generation to approval for payment.
During statement creation, REAP must be enabled for the statement to be eligible for auto-approval. This is done by simply selecting a check box in the statement generation tool. If a statement is eligible for auto-approval using REAP, it is validated to ensure that the data meets the SOX-approved control criteria.

Before REAP was connected to the payment system, we ran REAP on all the statements that were eligible, and Finance Operations reviewed the results until it was confident in the reconciliation accuracy and no recorded anomalies were being approved for payment. After the team was comfortable automatically approving transactions, it reduced risk by setting a maximum dollar amount threshold that made more than half of statements eligible for auto-approval using REAP.

As a statement is auto-approved, a REAP Approved Snapshot Report is generated (Figure 4), providing analysts with a high-level view of what was done to approve the statement as well as a summary of why it passed reconciliation.

### REAP Approved Snapshot Report

<table>
<thead>
<tr>
<th>Statement Information</th>
<th>Approval Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract ID:</td>
<td>Royalty Earned Amount In Contract: 6255.38</td>
</tr>
<tr>
<td>Contract Description:</td>
<td>Royalty Earned Amount In USD: 6255.38</td>
</tr>
<tr>
<td>Reporting Period:</td>
<td>Payment Amount: 6255.38</td>
</tr>
<tr>
<td>Payee ID:</td>
<td>Payment Currency: USD</td>
</tr>
<tr>
<td>Payee Name:</td>
<td>Status: REAP Approved</td>
</tr>
<tr>
<td>Silo:</td>
<td>Approval Date/Time: REAP Approved/Change Date/Time</td>
</tr>
</tbody>
</table>

#### Detailed Logs

**Recon Rules**

- Quantity in Royalty Statement as % of Total Quantity: ≤ SOX-approved threshold
  - Observed Value: 98%
- Variance Threshold (QTV) between Royalty Statement and 3rd Independent Source: ≤ SOX-approved threshold
  - Observed Value: 2%
- OR Total Quantity Difference between Royalty Statement and 3rd Independent Source: ≤ SOX-approved threshold
  - Observed Value: 1%
- Royalty Earned Amount is below SOX-approved threshold
  - Observed Value: Royalty Earned Amount in USD = 6255.38

**Recon Results**

- Passed
- RAID Results: Green

**Product Units for Royalty Statement and 3rd Independent Source**

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Product Name</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>Game_Name Game_Name_123456</td>
<td>542</td>
</tr>
<tr>
<td>1234567</td>
<td>Game_Name Game_Name_1234567</td>
<td>8</td>
</tr>
</tbody>
</table>

**Figure 4.** An example of the REAP Approved Snapshot Report available for view in the statement portal.

### Saving time and improving accuracy

Using the new workflow, we have been able to improve the speed and accuracy of anomaly detection and reconciliation. We are seeing reconciliation result accuracy greater than 95 percent, significantly reducing the number of statements that are routed through the manual approval workflow to less than five percent. The amount of time it now takes to process a royalty statement is down from an average of five days to four hours, as illustrated in Figure 5.
**Best practices**

We made a few decisions along the way that we felt contributed to the overall success of the project. The following best practices are based on our experience, and we hope they might give you something to think about if you are considering using machine learning or implementing automation, particularly around a financial process.

**Engage stakeholders and sponsors from the start**

One important contribution in the success of this project was the sponsorship and adoption of a digital transformation mindset. Leadership in Finance Operations and in CSEO endorsed and spearheaded the initiative, and their sponsorship helped engage the right stakeholders from audit, compliance, business development, and other groups. After the stakeholders were on board, it was easier to gain their support while we were working to turn this transformative idea into reality.

**Combine domain knowledge and analytics for a good outcome**

We had to start our work by partnering with the people in the business who truly understood the scenarios that we were trying to solve. Even before we began developing the solution, we had to consult with experts across Finance Operations to understand what it took to perform its manual reconciliations and how it detected anomalies. Team members helped us define what factors could make something look like an anomaly when it wasn’t actually an outlier after considering larger trends. On our end, we consulted with a data scientist to help us understand the results we were getting in RAD as well as to help us “reduce the noise” or refine our results by tuning the dials on our formulas to detect anomalies.

**Pilot in production**

We piloted much of our work in production, running RAD, three-way reconciliation, and REAP as if they were live, but not connecting REAP to the actual payment system. This gave the analysts in Finance Operations an opportunity to run their manual processes and compare them to the results we were generating using automation and machine learning. This was important because it allowed us to move forward slowly while building automation around a SOX process that requires a lot of oversight and operational controls. We needed to get it right, building the guardrails as we went, and they needed to build the confidence required to trust the results.

**What’s next**

As we move forward, we are looking at the success that we have had in using machine learning for anomaly detection and the ways automation can improve how repeatable, traditionally manual tasks are performed. We are currently planning to include a higher percentage of total statements on the “happy path” and have defined the criteria necessary to moving up to higher payment thresholds. It requires that we continue to work closely with the business to address all of its SOX and operational controls.

Other groups at Microsoft are working on new, more advanced machine learning models—models that better understand context and relationships in comparative data. They will be better at not only detecting anomalies, but also in explaining them. We look forward to incorporating those models as we work to improve and evolve our royalties-processing solutions.

**For more information**

*Creating efficiencies in finance with Dynamics 365 and machine learning*

*Using AI and chatbots to simplify finance tools at Microsoft*
Microsoft Dynamics 365 and AI automate complex business processes and transactions

Driving Microsoft’s transformation with AI

Accelerate your business with the power of data and AI

Microsoft IT Showcase

microsoft.com/itsshowcase

© 2019 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.