



# HR/Payroll (Workday) Data Conversion Chronicles

---

February 2018, UW Campus Tech Talks

UNIVERSITY *of* WASHINGTON

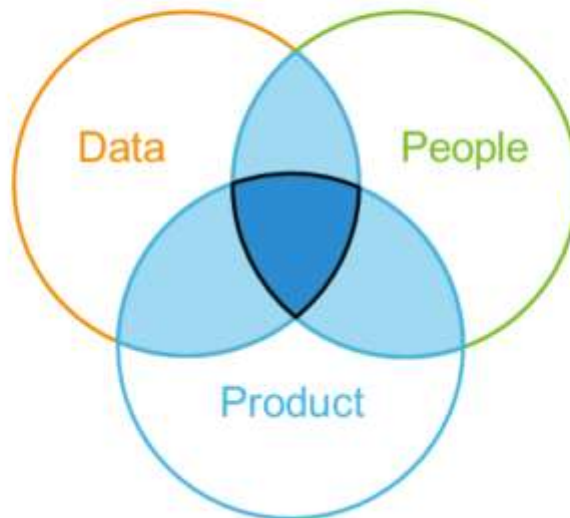


# Agenda

- ▶ Data Conversion – The Big Picture
- ▶ Save a Billion Dollars
- ▶ It's About the People
- ▶ It's About the Data
- ▶ The Product
- ▶ Behind the scenes

Required slide

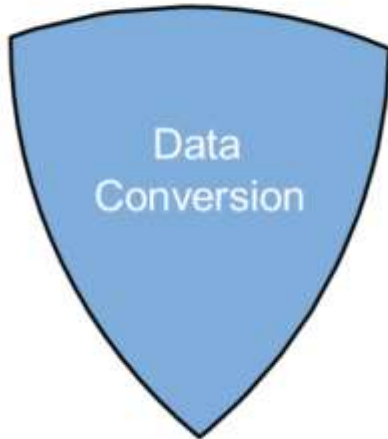
## The Big Picture



“Data conversion” is a misnomer. This implies a simple mapping of data fields from one system to another. In reality, transitioning from one system to another requires a much broader understanding of what data conversion is so that it can be done successfully.

Data conversion is the combination of raw data and the masterminds that can populate the final product with correct, consistent, and complete data. In other words, it’s the confluence of these subjects: Data, People, Product.

## Save a Billion Dollars



- Treat data conversion as a “shield.”
- Don’t pretend that data conversion is magic.
- Understand the data you have as well as the data that is “missing.”
- Give legacy functions/data their due.
- Document your findings.
- Automate your data conversions.

Data conversion done properly can save you a billion dollars. Examples: One branch of the military has undertaken a huge, multi-year ERP implementation effort. The complexity of converting data from multiple source systems, especially for a large, government entity cannot be understated. Just to dig out of the issues they’ve encountered, it will cost \$1.1 billion dollars. Similar story for the Washington State Community and Technical Colleges – not quite a billion dollars in play, but it’s been several years and other than 3 beta launch sites, the rest of the 30 colleges have yet to implement their chosen product (not Workday). Expensive.

Data conversion is a “shield” – if done properly, it will protect your transition to Workday. Data conversion transforms ideas into a hands-on experience and allows other teams to further their own work. If done improperly, data conversion poses the highest risk to a successful implementation – you won’t have a useable product if people don’t trust the data within.

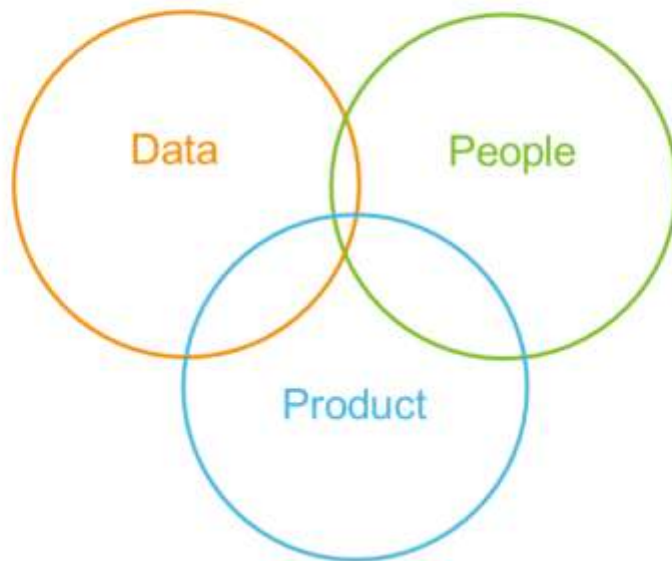
Obtain a deep understanding of your current data (or lack thereof), as well as current functions. It took years to create today’s legacy systems and they’ve been modified so much that they’re unrecognizable. People with historical knowledge of these systems may be long gone. You can’t Google an answer. Don’t minimize the need to understand the legacy systems before bringing them forward.

Document your findings about the legacy functions that will be transformed to

Workday.

Automate your data conversion. Make it repeatable. At the UW, we built 10 Workday prototypes, and 15 mini-prototypes. We treated data conversions like building and running a data warehouse, making the Extract, Transform, and Load processes as automated as possible.

## Initial Setup



Workday Confidential

The big picture, this time without the shield in the middle. On our project, there was a lot of emphasis on the Product and People intersection. Not enough emphasis on the Data and People intersection. The trick is to balance the focus of each area so that there's a strong shield in the middle.

## It's About the People



- Resourcing
- Hard skills
- Soft skills
- Historical knowledge
- Transition Team should be formalized

Don't underestimate the type and amount of work that data conversion is. You need all sorts of skillsets and knowledge in order to transition legacy systems to Workday. Some of the hats we wear on the Data Conversion team are: Systems analyst, business analyst, data modeler/designer, developer, facilitator, subject matter expert, project manager, QA/tester. Your SME's need to be both business and technically oriented. Some should be able to reverse engineer the legacy systems.

Transitioning from a 35 year old system is very different job than transitioning from a 10 year old system. The resourcing of your transition team needs to account for complexity and ambiguity – there's a lot of catch-up that probably need to be done both knowledge-wise and system-wise.

It turns out that soft skills and fortitude are just as important when doing data conversion. The hours were long and pressure packed. When the going got tough, we doubled-down and saw things through.

There should be a Transition Team on your project. This would reduce the silo's that we experienced – most people were focused on future state, rather than bringing the current state forward. The Data Conversion team filled all the roles of a Transition Team, however, that was by accident and not planned. The Transition Team works alongside the those focused on future state, alongside those focused on current

state, and they write the specifications for data conversion, and integrations. They are also an asset that is portable from project to project.



## It's About the Data



- Data sources—lots of them
- Data conversion, a.k.a. system-to-system conversion
- Missing data sources
- Dirty data
- Ambiguous requirements for conversion

At the UW, we had 64 data sources, 12 of them were formal systems, and 52 of them were collected or created out of thin air. At the outset, we thought there might be no more than 30 sources, but as Workday design continued, we ended up with 64. In addition to our 35 year old HR/Payroll system, the Medical Centers had Lawson and Kronos, and we had homegrown systems such as Academic Personnel and HRIS. Each of these system owners contributed their talent and knowledge and worked alongside the formal data conversion team. The effort was really a system-to-system conversion, not just data conversion.

**Missing Data:** If data to populate Workday for a particular function is not available in a centralized system, we called this missing data. Our functional teams were generally responsible for acquiring this missing data, as they knew how it needed to be used in Workday. The problem was that these 52 sources needed to be collected with each prototype.

**Takeaway:** If there's more than 100 rows of data to collect, automate it, don't try to do this with a souped up spreadsheet. Supervisory Org was a new concept for our departments, and it was collected for 60,000 positions via spreadsheet (never mind that we converted 70,000 positions). Even with a lot of instructions and hand-holding, each department was free to interpret Supervisory Orgs as they wished. It becomes a mess for data conversion.

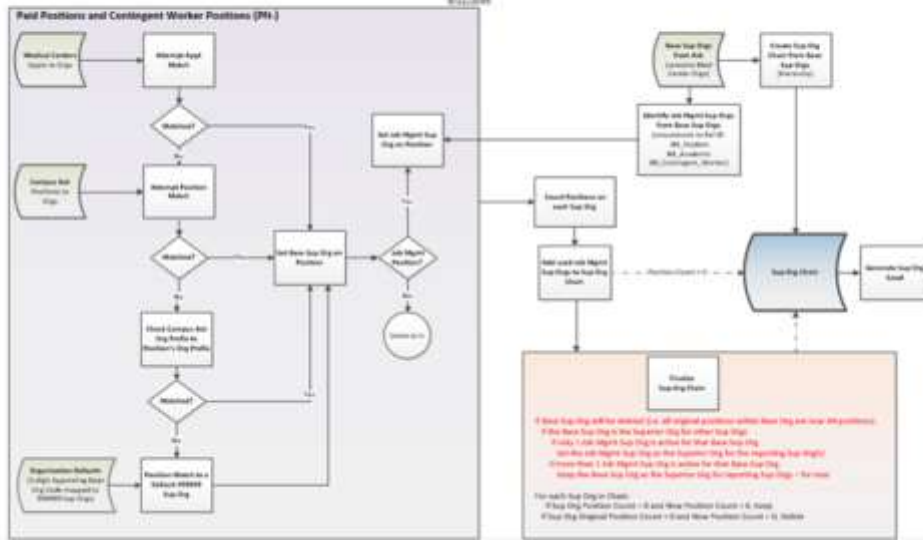
Even data that is in a legacy system can have issues. An example is Hire Date. There were at least 5 permutations of this date, and just as many systems that had their own version of Hire Date. Original Hire Date? Continuous Employment Date? Permanent Position Hire Date? In Higher Ed, faculty come and go, some may return in 3 months, others may not. Fortunately, when there was this type of ambiguity, the interested functional teams came to decisions together as to what was the master Hire Date, and also when to override that if another date in what circumstances. This turns into Transformation logic.

There will always be data gaps – something missing in the old systems that are needed for the future, or something not quite done the same in the old, and no programmatic way to make it happen for the future system. We dealt with this by taking a snapshot of the legacy data at a point in time, and generating Conversion Exception Logs with each prototype. The functional teams could then work with the departments to cleanse data at the source.

Ambiguous requirements. When the design of future state is still in process, as it will be until go-live, data conversion is not finalized. To deal with this, we modeled and created a flexible crossover system that pulled in the 64 data sources, did the thousands of transformations in a way that data integrity was maintained, and generated 225 data files for our Workday consultants to load. When requirement changes came along, we could spend our time focusing on getting the transformations programmed, rather than spending time dealing with infrastructure. As an example, Positions were a central entity in the crossover system. 2/3's of the way into the project, it was decided that the UW shall have Job Management positions, in addition to the existing Position Management positions. Because we had invested in automating the conversion infrastructure, we could be fairly nimble at altering the the crossover system.

# It's About the Data

F50 Supervisory Organization with Job Management – Process Flow



Not expected to read this slide – it's an example of one of the many conversion processes that was built out and communicated to others.

# Product



- Recognize the complexity
- Flexible crossover system
- Testing
- Prototypes
- Asset development

There's a greater chance of a successful Workday implementation when the data conversion complexity is recognized and focused on.

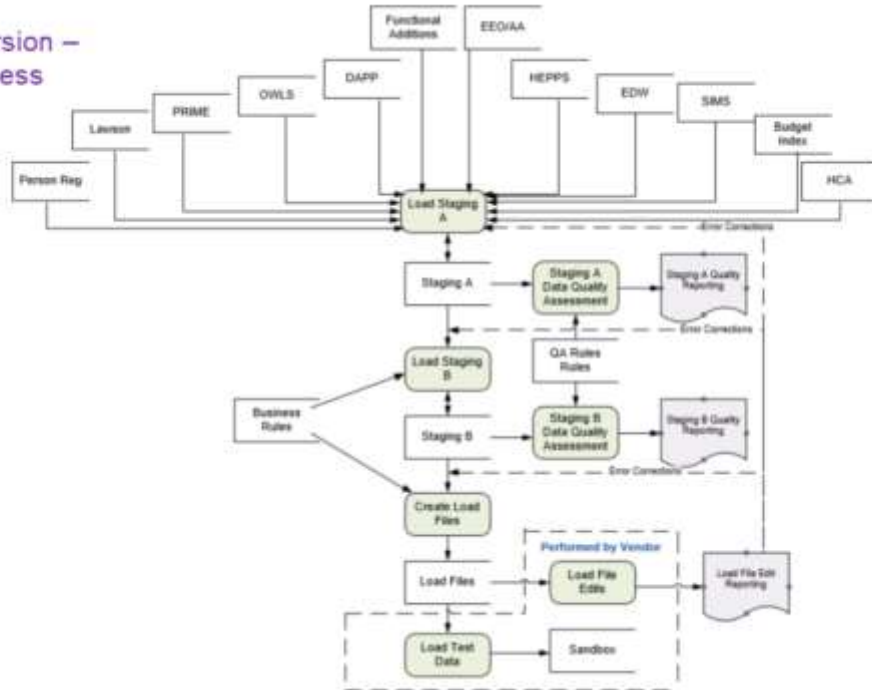
Make things easier: Build your conversion toolset – a crossover system that automates the ETL for data conversions. Automate your large data collection efforts.

Obtain a test tenant solely for conversion testing. Your first prototype builds will each take multiple weeks to complete. Procuring a test tenant allows you to vet new data transformations, eliminating the need to wait months between prototype builds and alleviates the pressure of fixing something new when there are deadlines to meet.

Plan for several Prototype builds. Automate as much of your data load file generation process as much as possible, so that you can create fresh data in a prototype at anytime.

You are building people and tool assets. Leverage them on future modernization projects.

## Data Conversion – Logical Process



Logical flow of the ETL (Extract/Transfer/Load) process, similar to a data warehousing process. This made generating Workday-ready data files a repeatable process.

# Conversion Data Dictionary Sample

AcadEM File	Variable Field Name	Variable-Defined Field Description	SQL Specific Field Information	Field Name in Load File	Concept Converted Value	Conversion Logic	Legacy System - Table	Legacy System - Field Name	
Hire Employees	Employee	The Employee ID or Contingent Worker ID for the worker.	-	EmployeeID	-	Employee ID Number. Only hire employees to their initial primary position where Employment Type is not V, and is not a reserve officer position. Other positions will be added.	HEPPS	HEPPS_Assessment	Dw_EE
Hire Employees	Applicant	The ID is used as a unique identifier for Applicants.	Applicant ID is the same as Employee ID	ApplicantReferenceID	-	HEPPS Employee ID Number	HEPPS	HEPPS_Assessment	Dw_EE
Hire Employees	Employee Type	Returns the type of employee for the hire.	All Employee Types except for Contingent Workers and Academic Affiliates are defined through Hire Employees. Contingent Workers are hired using the Contingent Contingent Worker process. Academic Affiliates are created through the Acad Academic Appointment job.	EmployeeTypeID	-	See appendix A-2 for logic.	DAPP HEPPS FSA	DAPP Job Code HEPPS Appointment HEPPS Distribution HEPPS Employment Status Academic Only/Cont Contingent Worker/Supervisor	EmployeeType Dw_APT_EEE_CD Dw_APT_JOB_CLASS Dw_D01_JOB_TYPE_CD Dw_FULL_PART_CD
Hire Employees	Hire Reason	The hire reason for this business process.	Benefits for no position job with different Reason: Hire Surplus (for hiring a surviving spouse of a deceased employee). Retirement (for employees that are retiring and being hired into sup org for their record data feed into an employment relationship with the University. Includes regular and Fixed-term staff, temporary, academic, and student employees. Initial Contingent Workers	HireReasonID	-	1 if position ID starts with 'V' then Hire_Employee_Benefits_Other_Retirement 2 if position ID starts with 'C' then Hire_Employee_Benefits_Other_Survivor 3 Otherwise Conversion	-	-	-
Hire Employees	Hire Date	The most recent hire date for a worker.	Most recent date feed into an employment relationship with the University. Includes regular and Fixed-term staff, temporary, academic, and student employees. Initial Contingent Workers	HireDate	-	See appendix A-2 for more on position generation process See appendix A-5	HEPPS	HEPPS_Employment_Status HEPPS_Assessment	Dw_EE_LATEST_HIRE Dw_APT_B04_DT
Hire Employees	End Employment Date	The end employment date for a Fixed-term employee's contract.	Entered for employees with predefined end date such as fixed-term staff.	EndEmploymentDate	-	1 if HEPPS job code is an Academic Job Code Crosswalk. Refer to appendix A-2 2 if job code is HEPPS Employment Status - Temporary and	DAPP HEPPS FSA	DAPP Job Code HEPPS Appointment HEPPS Security Date	EmployeeType Dw_APT_EEE_CD Dw_APT_JOB_CLASS

Sampling the Conversion Data Dictionary, built from SQL code into English so that a variety of non-technical people could understand and sign off on the data transformations taking place. The build of the Data Dictionary - a step in the business requirements gathering process - should really be done up front if possible.

## Code Sample

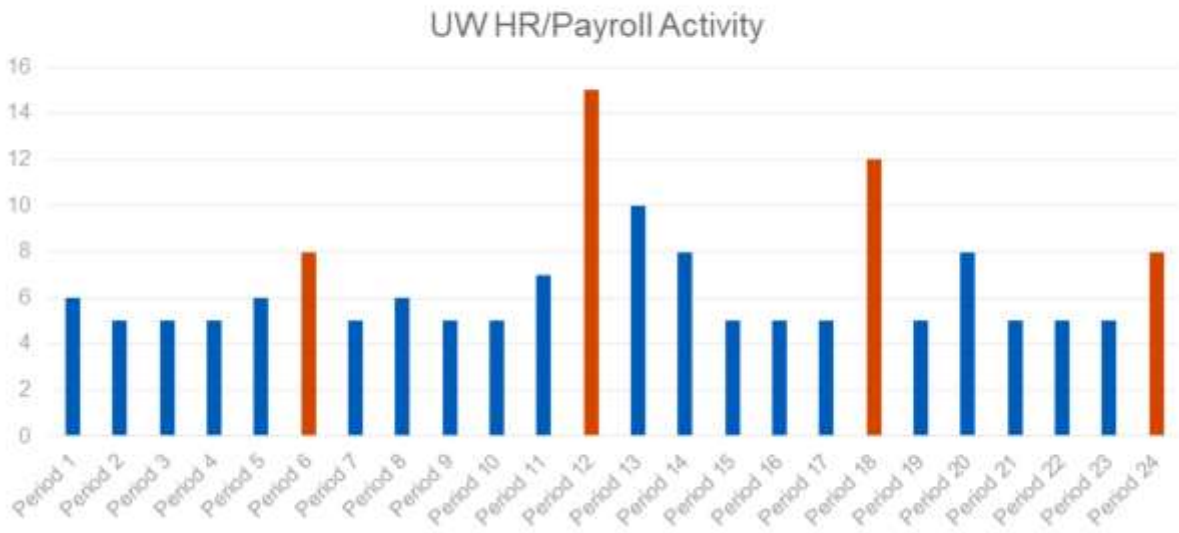
```
select ROW_NUMBER() over(partition by a.SID, d4.RangeNumber order by a.SID, PrimaryApptInd desc) OrigOrder
  ,a.PrimaryApptInd PrimaryApptInd
  ,case when a.MDEmployeeTypeDesc = 'Regular' then 1
        when a.MDEmployeeTypeDesc = 'Regular_Practice_Plan' then 2
        when a.MDEmployeeTypeDesc = 'Fixed_Term' then 3
        when a.MDEmployeeTypeDesc = 'Fixed_Term_Practice_Plan' then 4
        when a.MDEmployeeTypeDesc = 'Fixed_Duration_Leave_of_Absence' then 5
        when a.MDEmployeeTypeDesc = 'Fixed_Duration_Project' then 6
        when a.MDEmployeeTypeDesc = 'Temporary' then 7
        when a.MDEmployeeTypeDesc = 'Regular_Paid_Direct' then 8
        when a.MDEmployeeTypeDesc = 'Fixed_Term_Paid_Direct' then 9
        when a.MDEmployeeTypeDesc = 'Fixed_Term-6_Month_Summer' then 10
        else 99 end EmployeeTypePrioritySeq1
  ,a.MaxFTE as MaxFTEPrioritySeq1 --- Current FTE?
  ,case when a.EmploymentType = 'F' then 1
        when a.EmploymentType = 'T' then 2
        when a.EmploymentType = 'I' then 3
        when a.EmploymentType = 'H' then 4
        when a.EmploymentType = 'B' then 5
        when a.EmploymentType = 'C' then 6
        when a.EmploymentType = 'P' then 7
        when a.EmploymentType = 'U' then 8
        when a.EmploymentType = 'S' and isnull(dapp.InsuranceEligible,'N') <> 'N' then 9 --- Stipend with Benefits
```

Determining which position is primary. Example of data conversion logic, choosing which Workday position is marked as primary.

There were thousands of data decisions made over the life of the project, and the bulk of those decisions made it into the data conversion logic.

If time allows: Screen shots of UW data conversion built tools.

# Business Activity Cycles

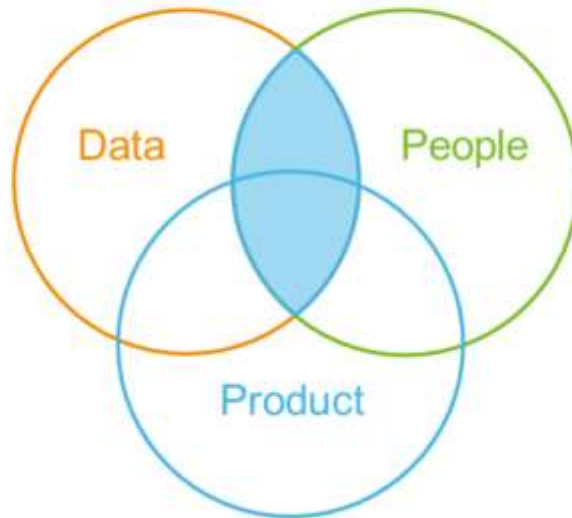


The original implementation date choice for HR/P was at the end of a tax year, after the last payroll process was run in December, a lower impact time to go-live. However, as the project pushed out, different implementation dates were chosen and “low-impact” couldn’t be the highest priority. Instead, Workday HR/P went live near Period 12, the busiest time in the yearly HR/Payroll business cycle at the UW (Spring Quarter ends, Summer Quarter starts, Biennium ends/starts). Lots of churn at this time, and that affected what data was available for converting, not a great thing.

Takeaway: know your business cycles and aim for implementing new systems during a quiet timeframe.



## It's About the Data and the People



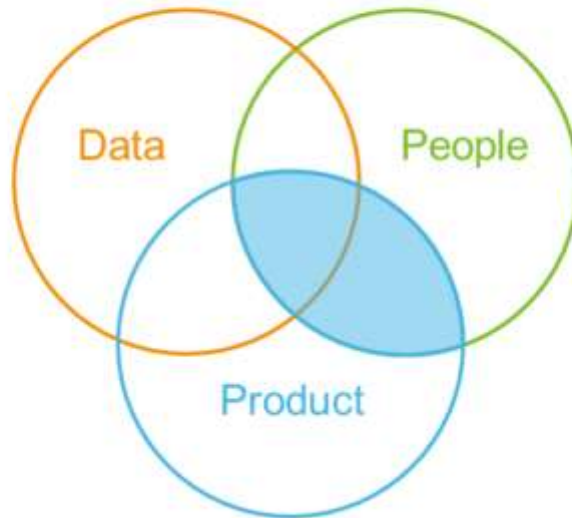
Data alone can't do much. With *only* data, the best you can do is simple moves from one place to another.

People without data on an implementation project is like working without facts.

The intersection of Data and People:

People enhance the value of data. They understand the who, what, where and why's of each data item, and can think in larger contexts on how individual data make a greater whole, and then can take the data in new directions.

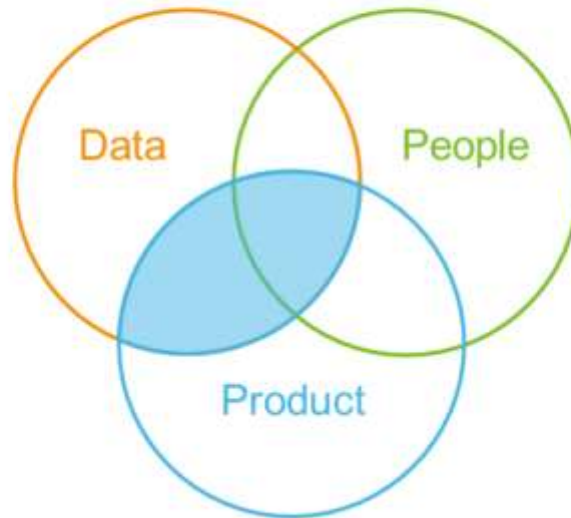
## It's About the People and the Product



The intersection of People and Product:

Workday is the “product”, what we’re aiming to transition to. With a product so vast, you’ll need people to understand the who, what, where and whys of Workday, from a configuration, front-end, and back-end perspectives.

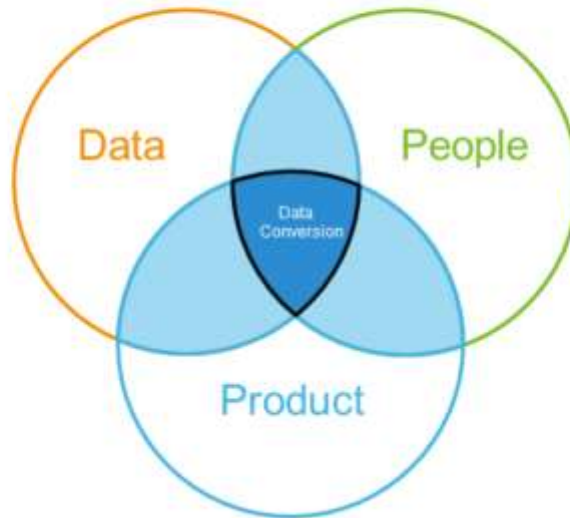
## It's About the Product and the Data



Intersection of Data and Product:

One without the other isn't useful. Your data, transformed into Workday data, will change lives (really). We built 10 major prototypes, pulling together Workday with transformed legacy data, providing a working tool that the rest of the project could rely on.

## Back to the Big Picture



### Summary:

Data conversion is the confluence of Data, People, Product. When the focus on all 3 is balanced (light blue sections), the data conversion “shield” (dark blue) is strong and you won’t have to rely on miracles for a successful Workday Implementation.



Thank you!

---

February 2018

UNIVERSITY *of* WASHINGTON

**HR/Payroll**  
MODERNIZATION