Microsoft

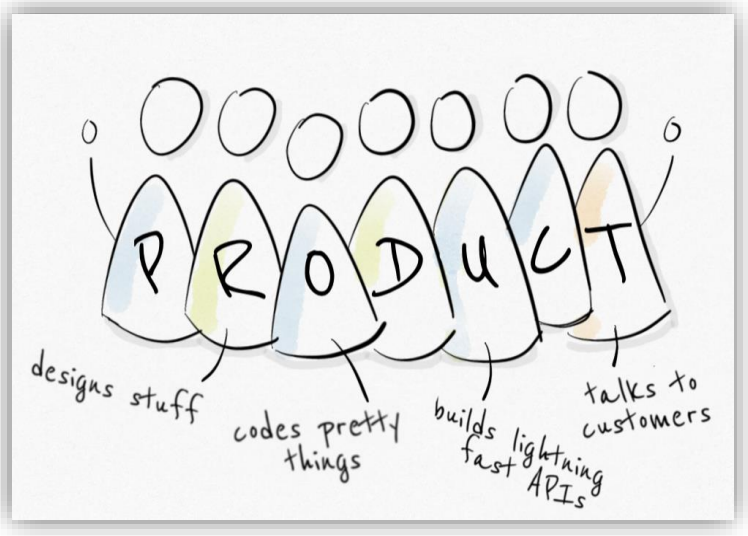
Using Design Thinking to Avoid Data Project Failures

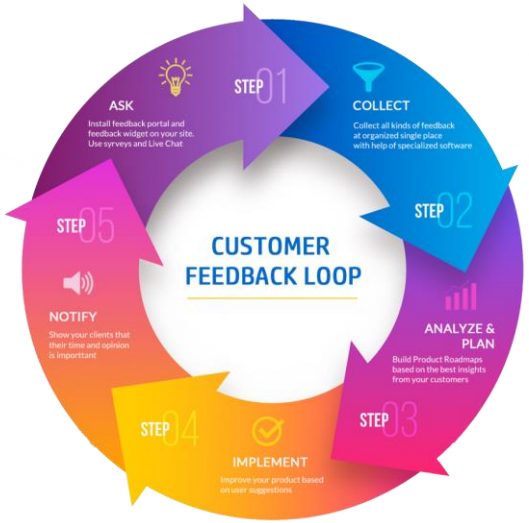
Maybe there's a better way than scrum and agile?

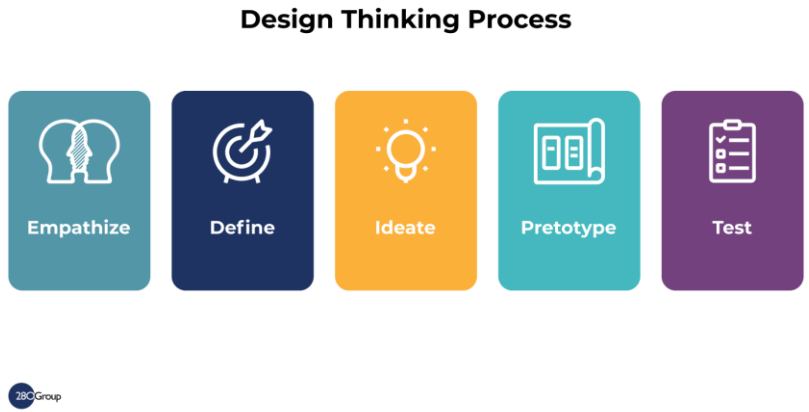
Dave Wentzel
davew@microsoft.com
linkedin.com/in/dwentzel

Once you know you're going to build an AI/ML/Analytics product, how should your team get started?

Getting started







Multidisciplinary Effort

- Empathetic
- All Personas Participate

Continuous Feedback

- Multiple, Progressive Rapid Prototypes
- Fail-fast, experiment, learn

Design Thinking

- Rapid Prototyping

Why do we do it this way?

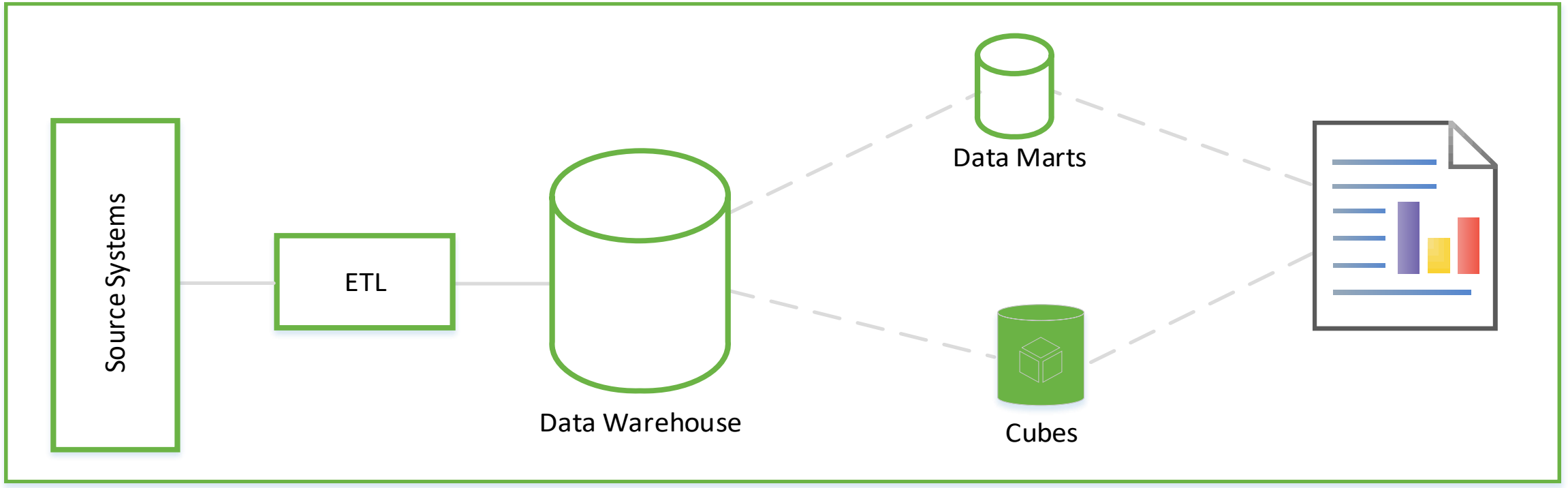
It Works!!!

1. We learned the hard way.
2. Data projects are risky with a high failure-rate
3. Constant customer communication clarifies what is important.
4. Data professionals have strong biases as to what we believe is easily understood.
5. Sometimes even a concept we think is simple like median or StdDev is hard to convey in a clear way.
6. Constant feedback loops should be built in to both the product and the project.

Why do data projects fail?

Why do Data Projects Fail? Legacy Thinking

The Philosophy: Model data » Transform data » Load data » *Understand* data

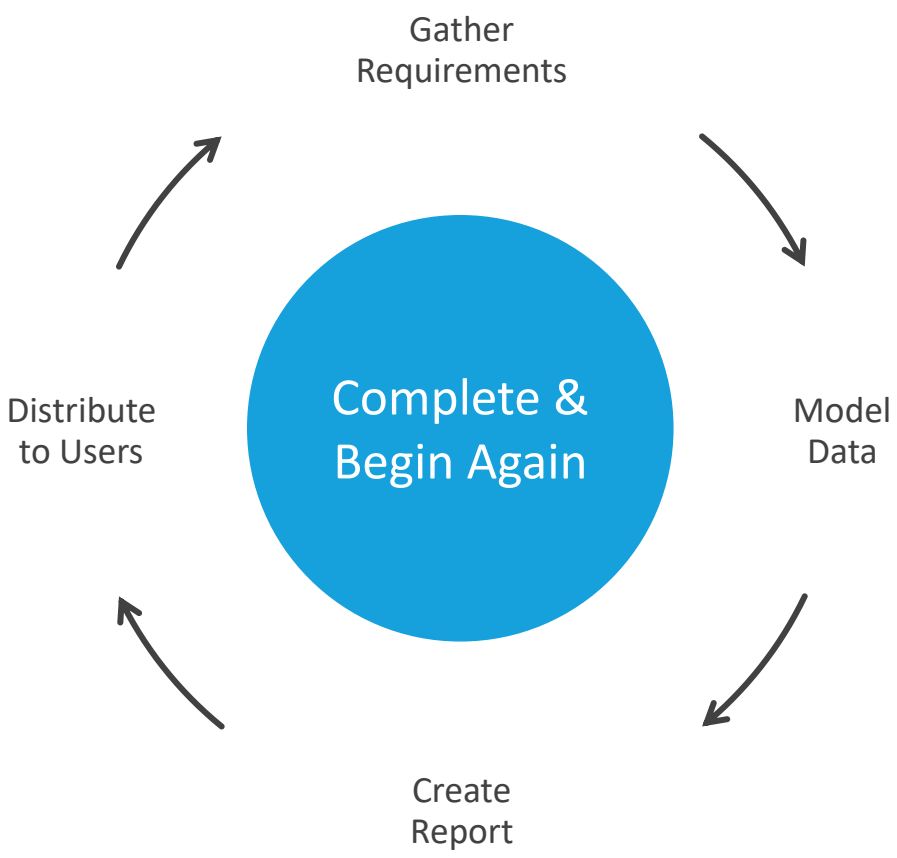


Data Projects have a high fail rate

Too much time is spent in:

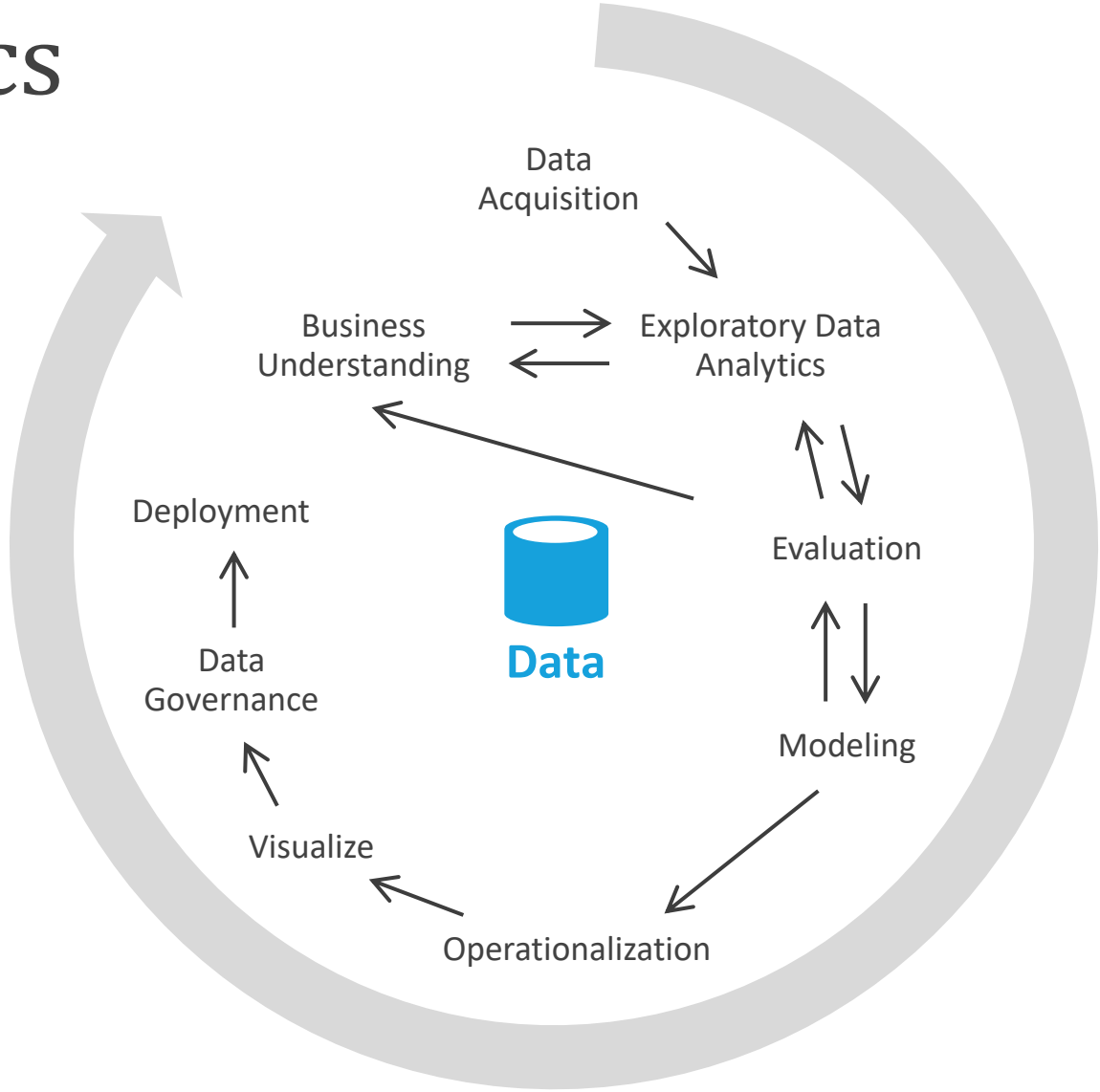
- Requirements gathering
- Data modeling
- ETL

Users only see the fruits of the endeavor after the reports are created



Exploratory Data Analytics

- A robust and well-proven methodology.
- Data science-like.
- Iterative.
- Stresses up-front understanding of data.
- Modeling is done later in the process (schema-on-read).
- ETL might not be needed



What is Design Thinking?

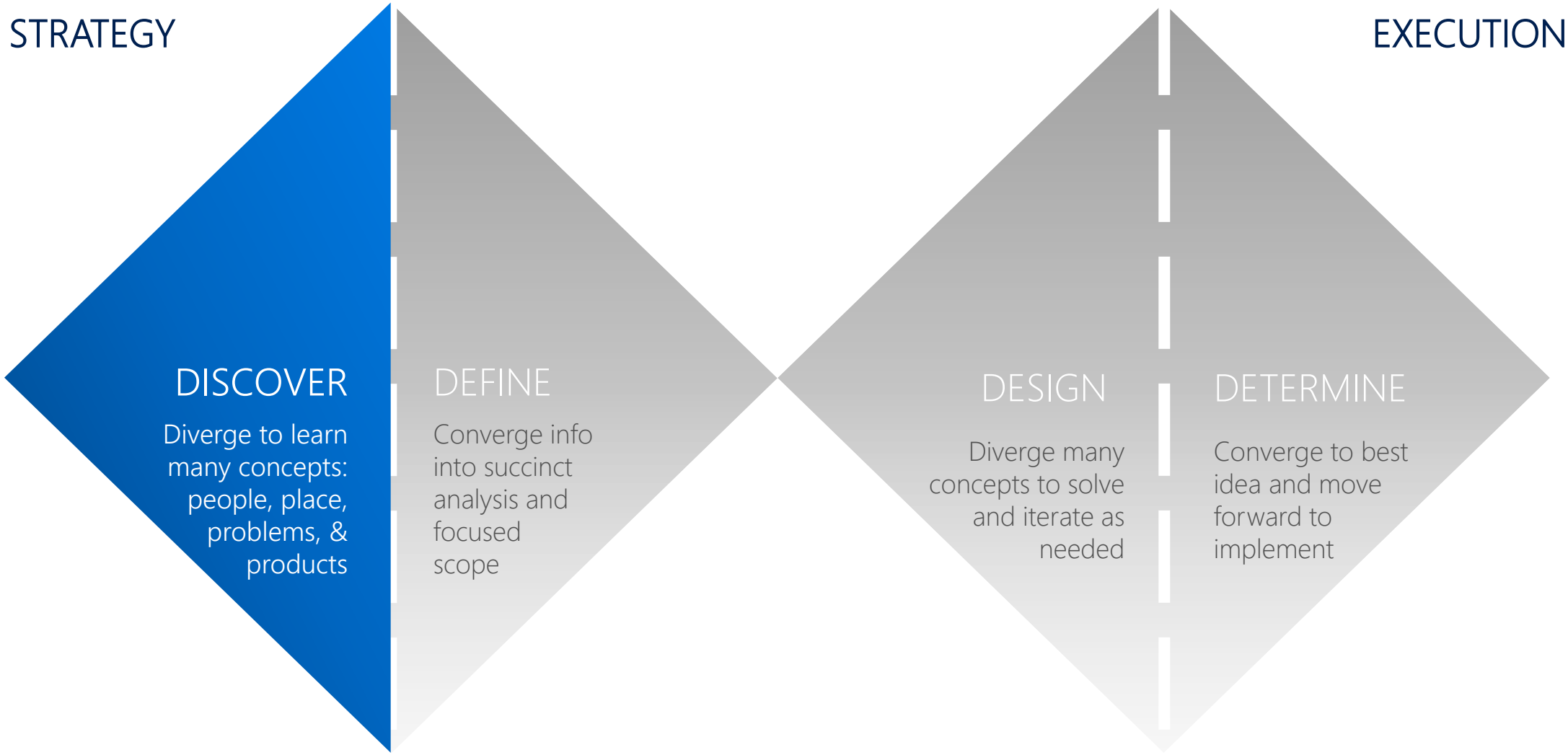

You're a
Superhero

Design Thinking: What is it?

- A gathering of stakeholders who have a common goal of identifying key business opportunities using human-centered, empathetic thinking.
  - Define/discuss current problems and uncover rapid prototyping possibilities that lead to faster business optimization
- The goal is tangible next steps and complete alignment on scope and use cases.
- Initial DT session is 1-2 days.
- On-going DT sessions can be as little as 2 hours.

STRATEGY

EXECUTION




Best way to have good ideas…
…is to have a lot of ideas!
#2 Build on top of others' ideas…
#1 First generate options…
As many as you can…
NO FEAR! Remember … YOU ARE CREATIVE
…but DON'T KILL others' ideas

When to Recommend a Design Thinking Session

IT has limited understanding or has questions about business needs.

IT wants to build or re-establish relationships with the business.

Analysts want to talk through problems without writing a requirements doc or user stories.

Business doesn't know what to prioritize or what is even feasible.

Traditional analytics projects are failing, and we want to try something different.

Business has an "analytics" strategy but needs help executing.


Be the girl on the right

Use Case Development…Start by Asking Interesting Questions

- What's the biggest challenge you face?
- What are you up against here?
- What causes the most frequent breakdowns?

Making a Business Case for Analytics

- You need a compelling reason to change anything, otherwise you get stuck in status quo
- Expose the problems and costs of status quo
- SHOW the end solution to sell the vision
- Communicate throughout the entire process
- Prototypes, Elevator Pitches, Storytelling

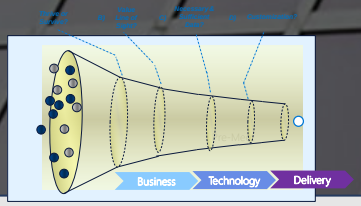
A business justification that you can sell to leadership

How to think about use cases

- Empathy: Ideate through the eyes of the end user experience– what's the persona of the end-user now and in the future?
- A use case is part of an overall journey – first step in a journey can be tied to aspirational and transformational future use cases
- Look for common themes or broad patterns – use cases can typically be classified by end user, by client focus area or by capability

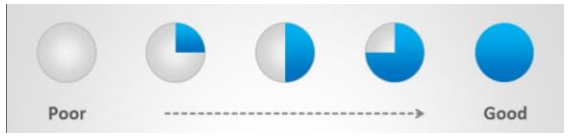
Identify broad patterns and end-users, think about a larger journey sequence

Data Estate



Ranking Use Case by Data Considerations				
Key Question	**Use Case (1st)**	**Use Case (2nd)**	**Use Case (3rd)**	**Use Case (4th)**
1. Relevant domain specific data available for the use case? 2. Outside domain data available for training sets (e.g. positive domain versus negative domain data)? 3. Where does the data currently reside/can it be readily accessed? 4. Format of data (i.e. file type, unstructured, etc)? 5. Quality of data (i.e. clean-up and suitability for training?) 6. Languages required?				

Use Case Prioritization Summary



Filter	Use Case 1	Use Case 2	Use Case 3
Transformational vs. Incremental			
Line of Sight to Value			
Data Estate			
Customization			
Road Map Trajectory			

Here’s how we start out our INITIAL DT session...

Recurring DT sessions can be much more streamlined

Design Thinking as a Project/Risk Management Tool

Problems with Scrum/Agile for Data Projects

1. Claims to “respond to change quickly”. Rarely does.
    1. “We can’t add that request to this sprint because sprint scope is locked. We can discuss this during the next Sprint Planning Meeting”
2. Too much focus on “points” to estimate work effort
    1. If it needs to be done, do it. Real, thoughtful estimates can be made _if required_.
3. Points & cards-on-the-wall are nebulous to a stakeholder. They want to see something tangible.
4. Focus on “burndown” instead of learning and changing direction.
5. Sprint durations are meaningless. We work until we run out of work, then we do another DT session (much closer to Kanban).
6. Velocity decreases at the sprint boundaries and just when we achieve good velocity we need to ramp down for Sprint closing.
7. While the goal is to have analysts, devs, and QA sit next to each other and work together, this rarely happens.

Solution: Lean, Kanban, and DT

1. DT replaces Sprint Planning
2. No need for cards-on-the-wall, “party planning poker”, user stories. Everyone knows what we are building and can work together as needed on the progressive rapid prototyping.
3. The goal isn’t to “claim points”, the goal is to progress the prototype.
4. No defined “sprint close” where the stakeholders review the sprint. Instead, we present the latest prototype and have honest conversations on progress/direction/priorities.

What every data PM should know?

1. Analytics projects are complex undertakings and require multidisciplinary efforts.
2. Embrace “fail-fast” and experimentation
3. Focus less on traditional metrics like velocity, burndown, points and focus on continuous feedback and TRULY responding to change.

Feedback Loops