



5 Keys to Graph Success

GEMINI

Johnny Lin

Co-founder & CEO

Dylan Hedberg

Sales Engineer

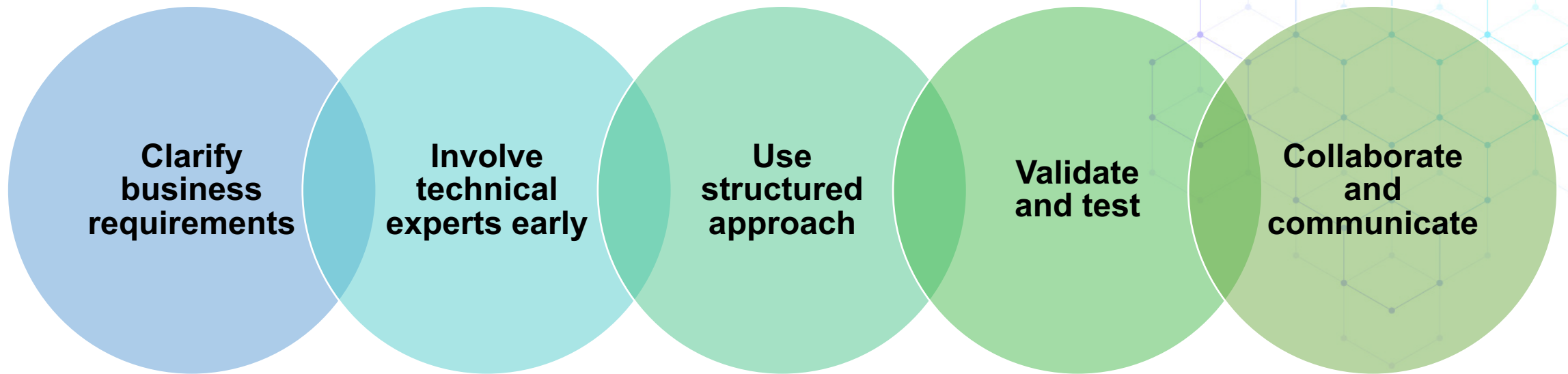
#1. Translating Business Requirements into Technical Requirements

Problems

Business requirements	High-level goals and objectives that a software application must meet to satisfy the needs of the business	Expressed in non-technical language and concepts.	Can be ambiguous and subject to change.
Technical requirements	The specific features and functions that the software must have to meet those business requirements	Requires a deep understanding of the underlying technology (graph) and systems.	Must be precise, detailed, and unambiguous

#1. Translating Business Requirements into Technical Requirements

How to fix this problem



#2. Ensuring Data Quality

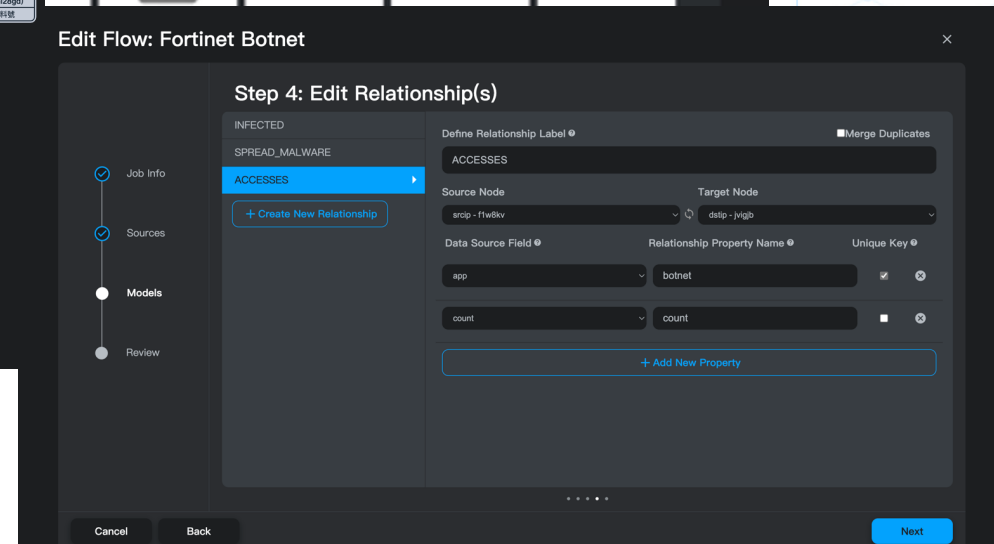
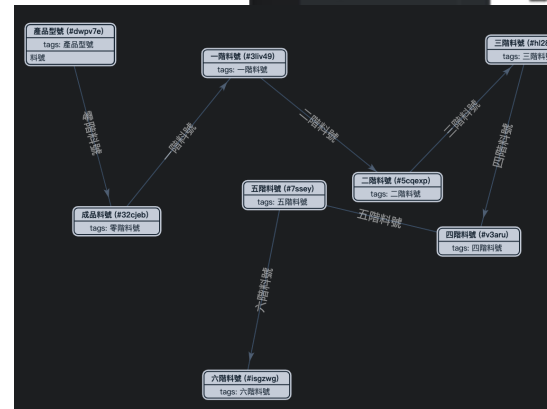
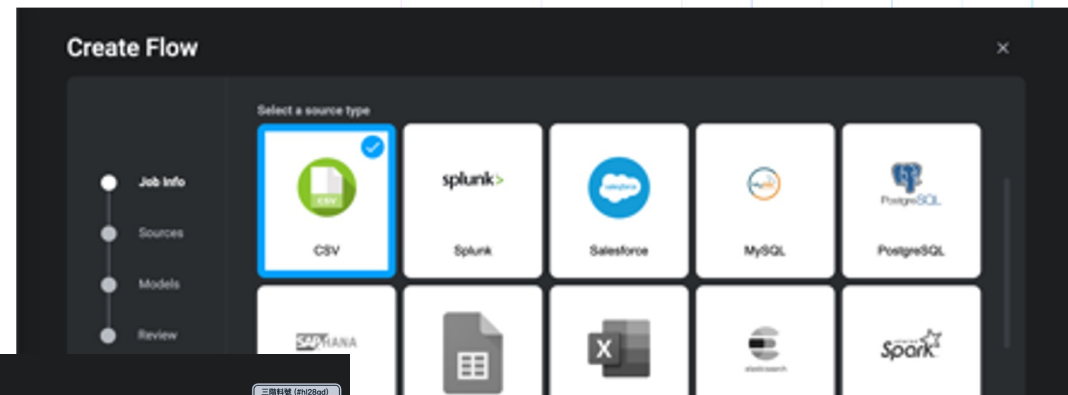
Getting access to data, how to connect, format and store it (quality)

Problem

Before you can query or visualize data you have to make sure it is normalized, ingested, and properly modeled.

How to fix:

- Normalize the data
- Ingest the data
- Identify entities
- Identify relationships
- Create nodes and edges
- Connect nodes with edges with proper data model



#3: Too Much Training Required

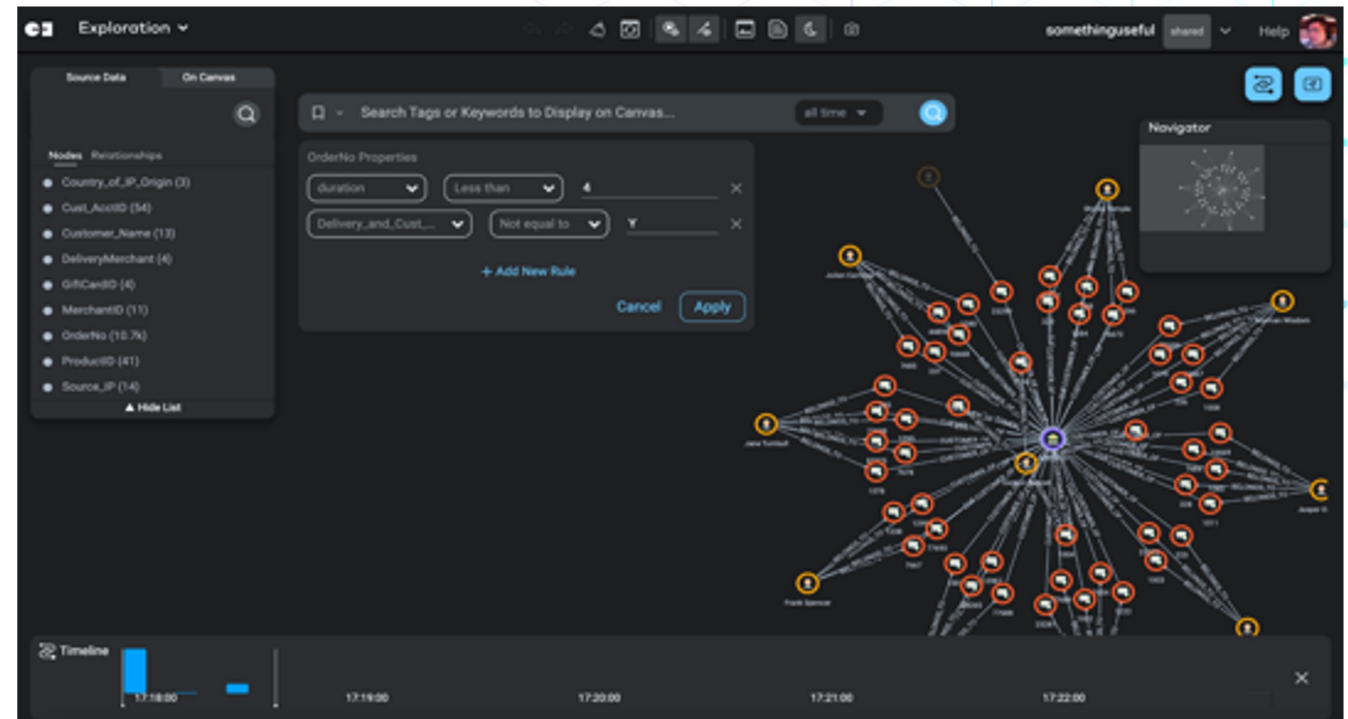
Specialized skills and query language required, long learning curve

Problem

If an application requires too much training or is too difficult to use, employees may avoid it altogether, leading to reduced productivity and decreased efficiency.

How to fix:

- Invest in the front-end development or intuitive tools
- Keep what's under the hood, under the hood



#4: Graphs are better shared

Enterprise collaboration and sharing

Problem

End users can't easily share views and analysis of graph data with their co-workers or colleagues

How to fix:

- Role-based access
- Knowledge creation and sharing
 - Snapshots
 - Sharable links
- Data enrichment
 - *bring in more data/context

The image displays a user interface for graph analysis with three overlapping windows:

- Save Snapshot:** A dialog box with a 'Name' field containing 'MS.DCERPC.NETAPI32.Buffer.Overflow Related', a 'Description' field with 'For the current investigations.', and 'Cancel' and 'Save' buttons.
- Snapshot Grid:** A main view showing a search bar and a grid of six snapshots. The snapshots are titled 'Power cord story', 'SKUs & Materials', 'APAC Part Anomaly', 'i found something useful', 'test', and 'APAC result'. Each snapshot includes a graph visualization and 'Created by' information.
- Share:** A dialog box showing a 'Link' field with a long URL, a 'read only' dropdown menu, a 'Copy' button, another 'read only' dropdown, a 'can explore' dropdown, and a 'Done' button.

#5: Generative AI makes graph more accessible than ever

AI-Powered, Context-Driven Decision Making

Problem

Enterprise-grade use of Generative AI requires a solid data pipeline and store as well as front-end to enable easy access for business users

How to fix

Human text input > Graph data output with context

*Graph databases excel at handling complex, interconnected data, making them ideal for generative AI training, unlike traditional relational databases that rely on tables and can struggle with complex relationships. In simpler terms, graph databases are like a flexible, efficient, and easy-to-understand web of information perfect for training AI models that need to understand complex connections.

By ChatGPT

