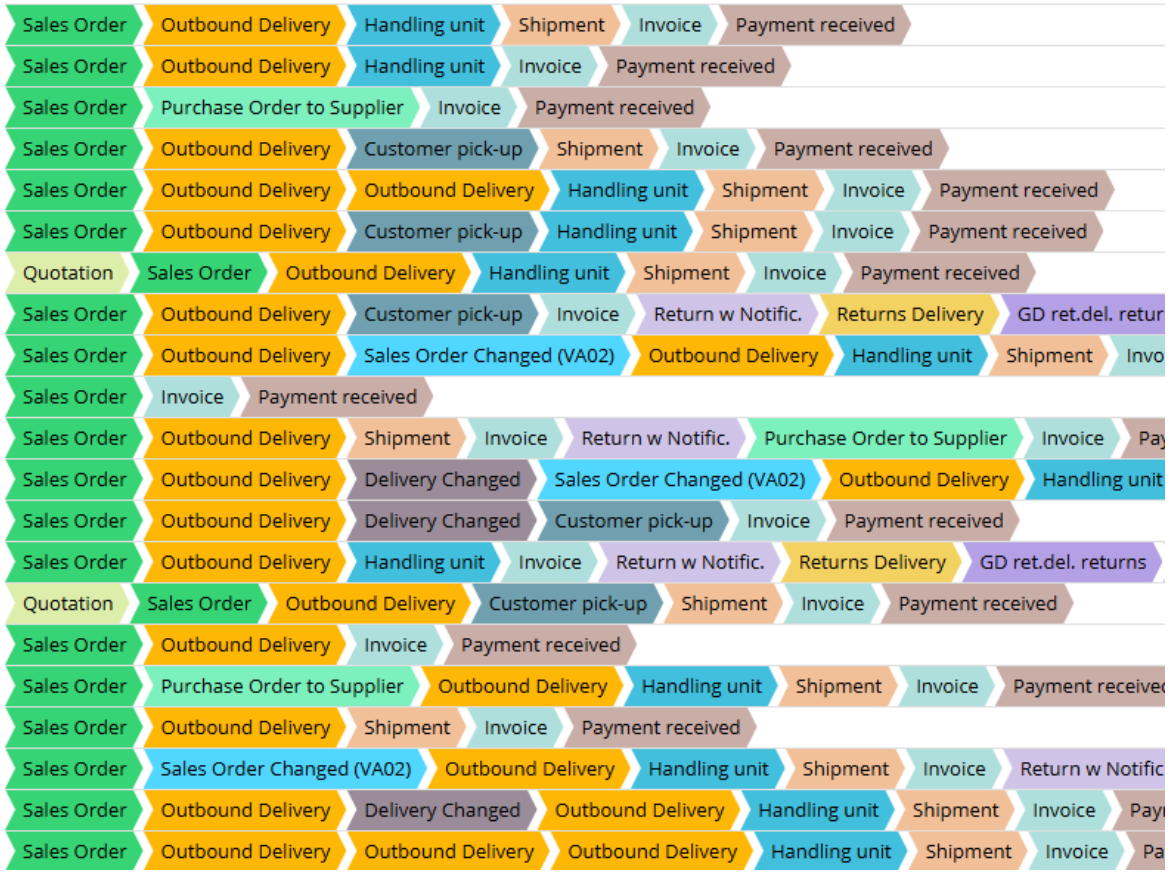
QPR ProcessAnalyzer 2025.1 – New features

Variations and other lists can be visualized with colors

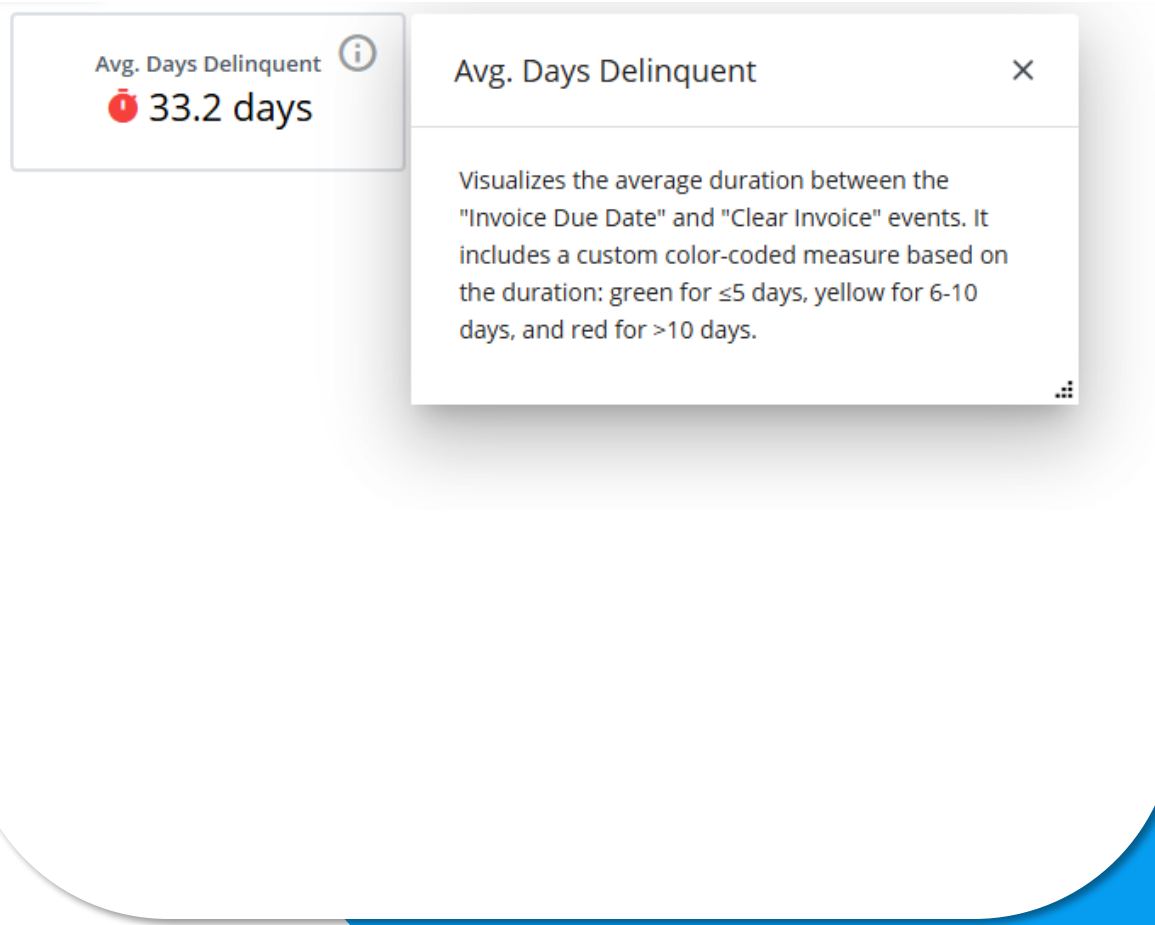
- “Visualize list items with colors” can be selected in measure/dimension/column settings to highlight list items with colors
  - Use cases: variations, paths, case/event attribute lists
- Colors are based on changeable color palette
- Items can be hovered to highlight same items
- Width for items can be set for better vertical alignment
  - When not set, items have variable widths based on text length
- Introduced new data type: List
  - "List separator" field is available for that type
  - List type can be set for custom expressions
- List item visualization is taken into use in presets showing variations and paths



Description text can be added to chart for additional explanation

- Charts and other dashboard components can have user-defined description to provide more information
  - When description is defined, “info” button appears which opens the text in a dialog
- Description field allows Markdown and basic HTML formatting
  - E.g., bolding, italic, headings, lists, links etc. can be used
  - Links in description will open to new tab
  - See examples from Snowflake presets

  markdownguide.org/basic-syntax
- User can interact with dashboard while description is shown
- AI Assistant uses chart description as prompt
- Snowflake presets in Process Discovery tabs have description defined



LlmComplete function to access Snowflake language models

- Added SQL expression function **LlmComplete** to call Snowflake Cortex language models
  - Can be used in chart custom expressions, so eventlog data can be provided as prompt
- Not all LLM's are available in all regions, so *Cross-region inference* is needed to use LLM's in other regions

https://docs.snowflake.com/en/user-guide/snowflake-cortex/cross-region-inference

- There are several LLM's available (e.g., Meta Llama, Mistral AI, Anthropic Claude) which vary in capability, cost, and latency
  - Using LLM's consumes Snowflake credits
  - Some LLM's seem quite slow
  - Some LLM's have short context window that doesn't work with long prompts
- OpenAIChatCompletion function provides similar functionality, except
  - It's not available in Snowflake Native App
  - It cannot be used in SQL expressions

wiki.onqpr.com/pa/index.php/SQL_Expressions#LLMComplete

Short syntax:

```
LlmComplete(
  "snowflake-arctic",
  Concat("Explain following: ", Column("Product Group"))
)
```

Long syntax:

```
Cast(
  LlmComplete(
    "llama3.1-405b",
    [
      #{
        "role": "system",
        "content": "Explain the given product group."
      },
      #{
        "role": "user",
        "content": Column("Product Group")
      }
    ],
    #{
      "temperature": 0
    }
  )["choices"][0]["messages"],
  "string"
)
```

Project-level Snowflake database and schema available as separate settings

- Added following settings to define project's Snowflake database and schema:
  - DatabaseNameInDataSource
  - SchemaNameInDataSource
- Easier to use than connection string as no secrets are needed, and current values can be seen
- Project-level connection string is still required if connecting to different Snowflake account or user is different
  - These are not available for Native App
- Also project-level warehouse requires connection string setup, but plan is to have model-level warehouses

wiki.onqpr.com/pa/index.php/QPR_ProcessAnalyzer_Project_Workspace#Project-level_Snowflake_Database_and_Schema

Set project's database and schema:

```
ProjectById(1).Modify(#{
  "DatabaseNameInDataSource": "My database",
  "SchemaNameInDataSource": "My schema"
});
```

See current value:

```
ProjectById(1).Configuration
  .TryGetValue("DefaultLocationInDataSource")
  .TryGetValue("Database") ?? null
```

Other improvements

- Added **CreateProject** function for creating projects using expression language
  - Can be called to parent project to create sub projects, or in generic context to create root projects
  - Needed for generic export-import functionality
  - Will be used to create example content for Native App
- Object-centric model filtering backend available

wiki.onqpr.com/pa/index.php/Filtering_in_QPR_ProcessAnalyzer_Queries#Object-centric_filter_rules

```
ProjectById(1).CreateProject(#{
  "Name": "My project",
  "Description": "My description",
  "DatabaseNameInDataSource": "My database",
  "SchemaNameInDataSource": "My schema"
})
```