So, You Want To Build A Dashboard…

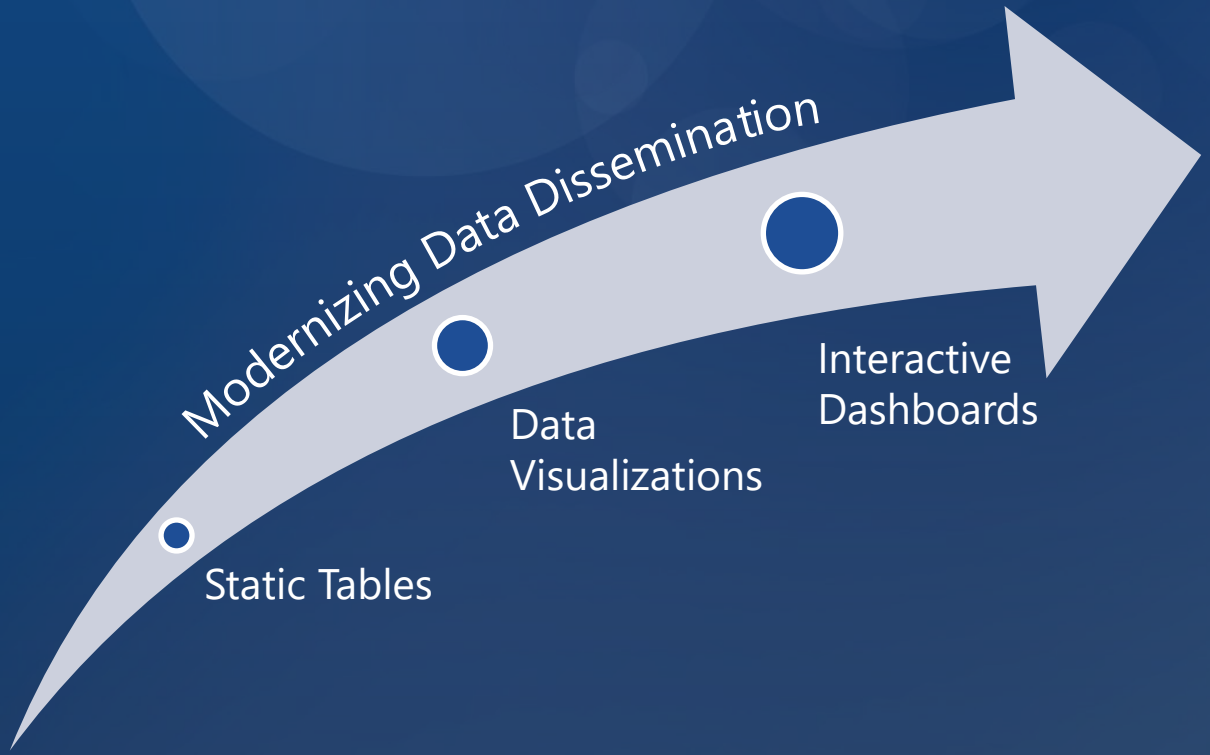
Choosing The Right Digital Dissemination Tool

Alex Harding

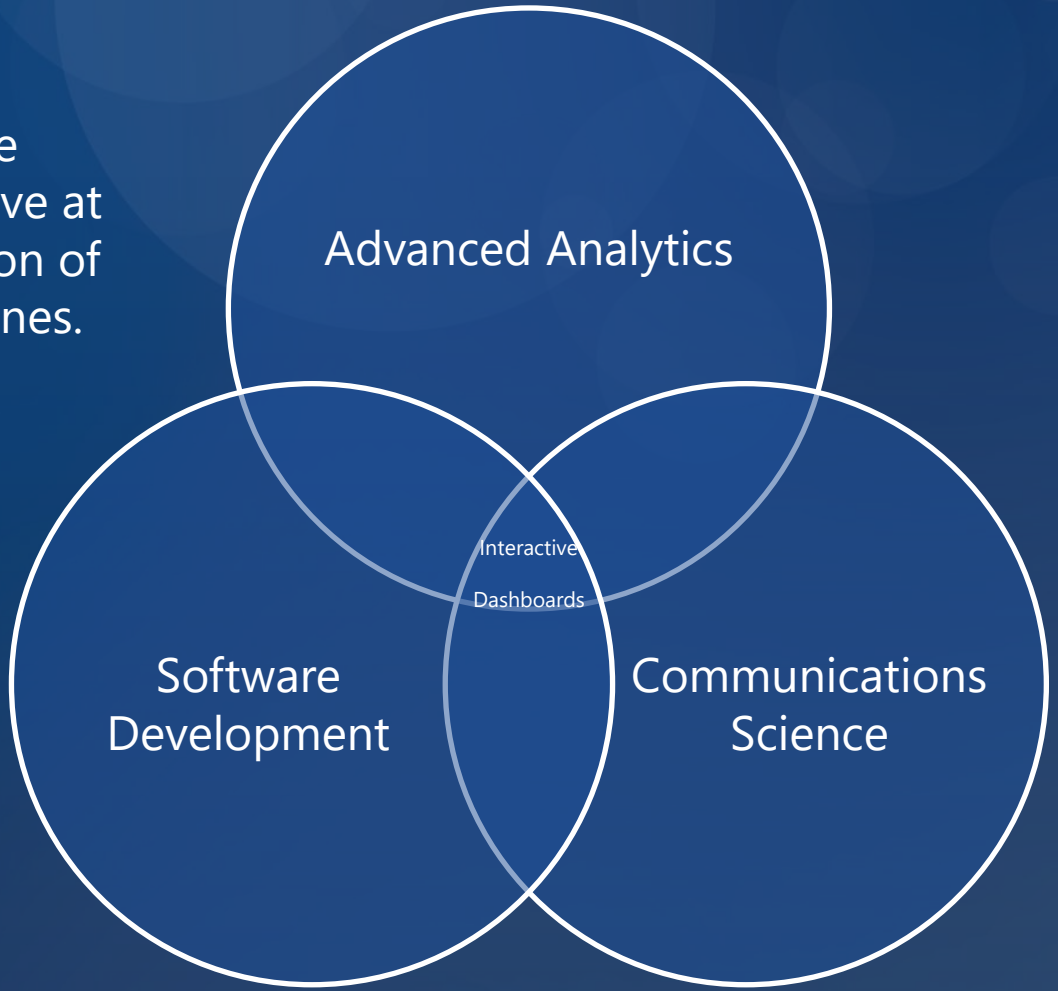
FCSM 2024

RTI INTERNATIONAL

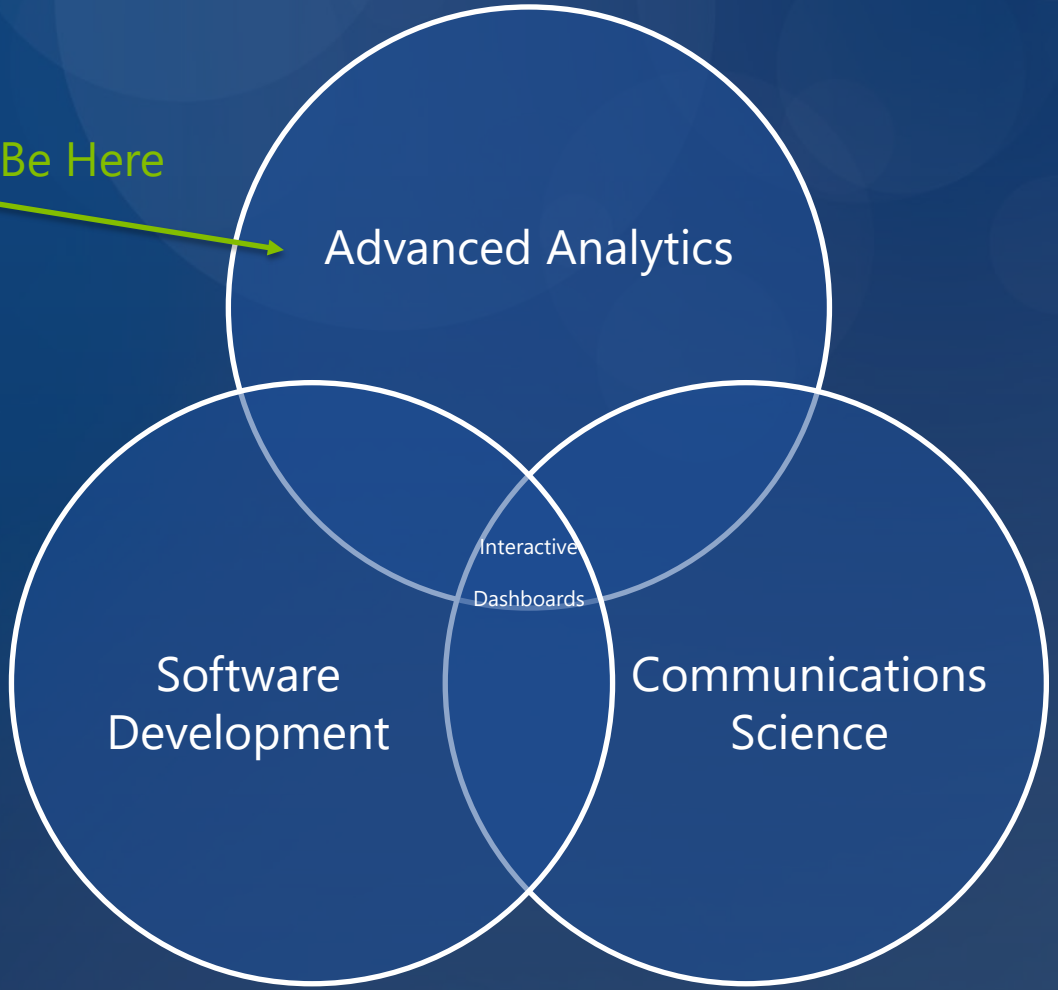
RTI International is a trade name of Research Triangle Institute. RTI and the RTI logo are U.S. registered trademarks of Research Triangle Institute.

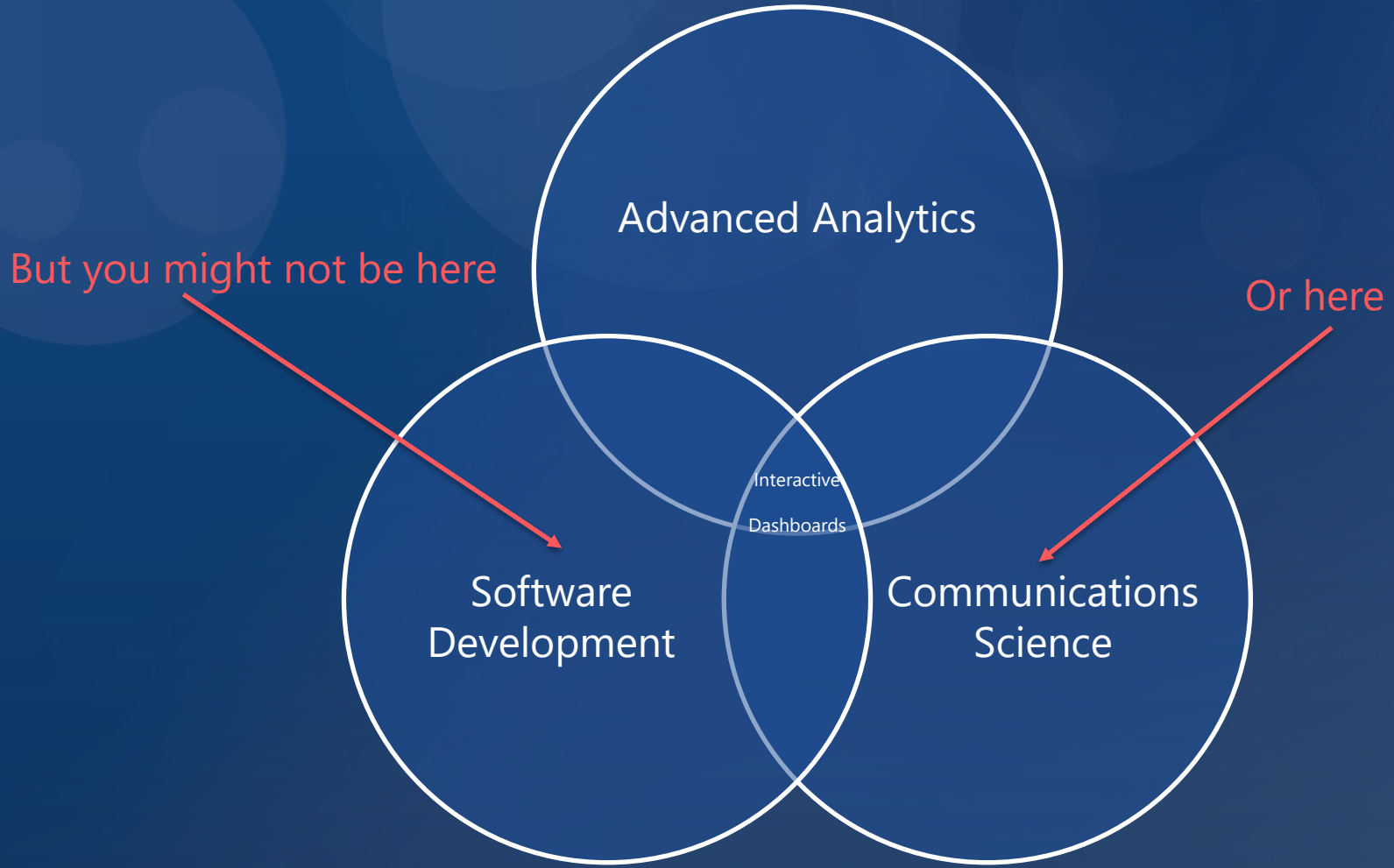

Modernizing Data Dissemination
Static Tables
Data Visualizations
Interactive Dashboards

Interactive Dashboards live at the intersection of three disciplines.

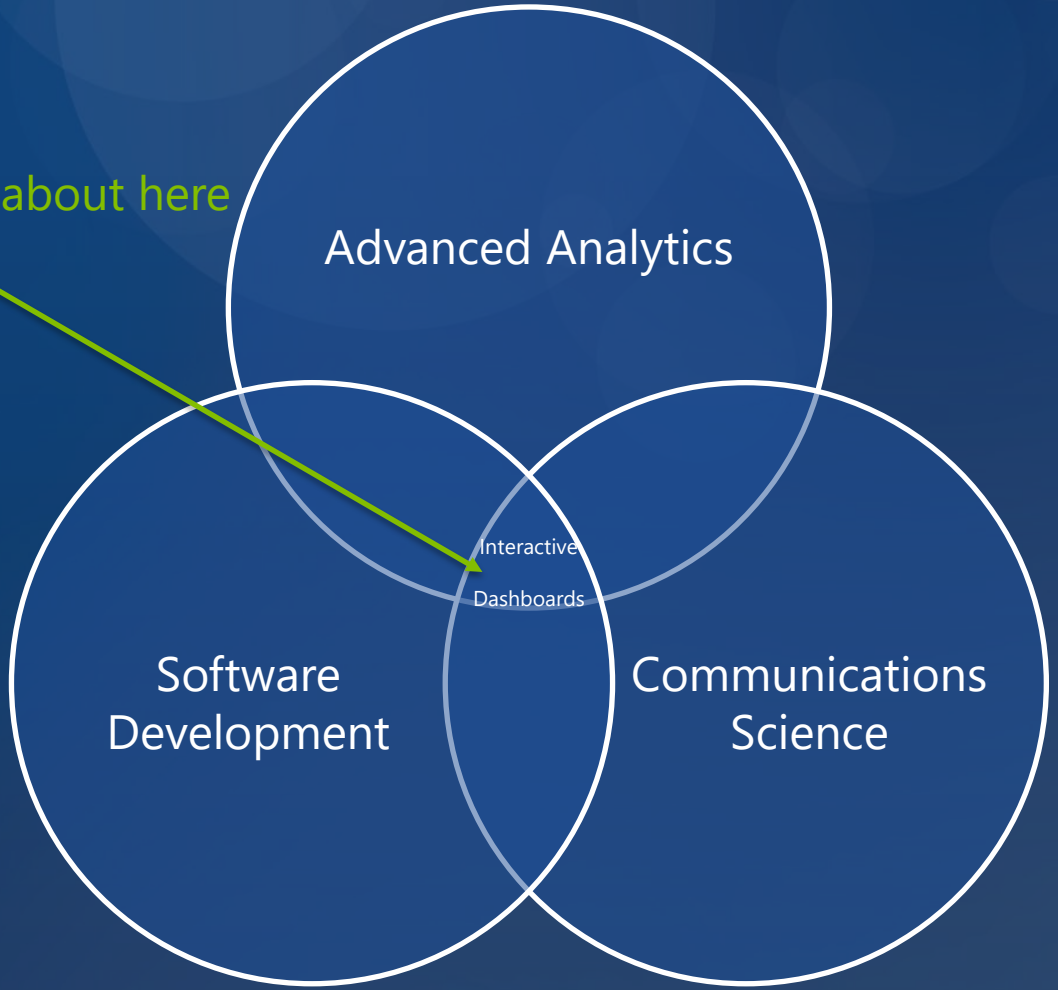


You Might Be Here



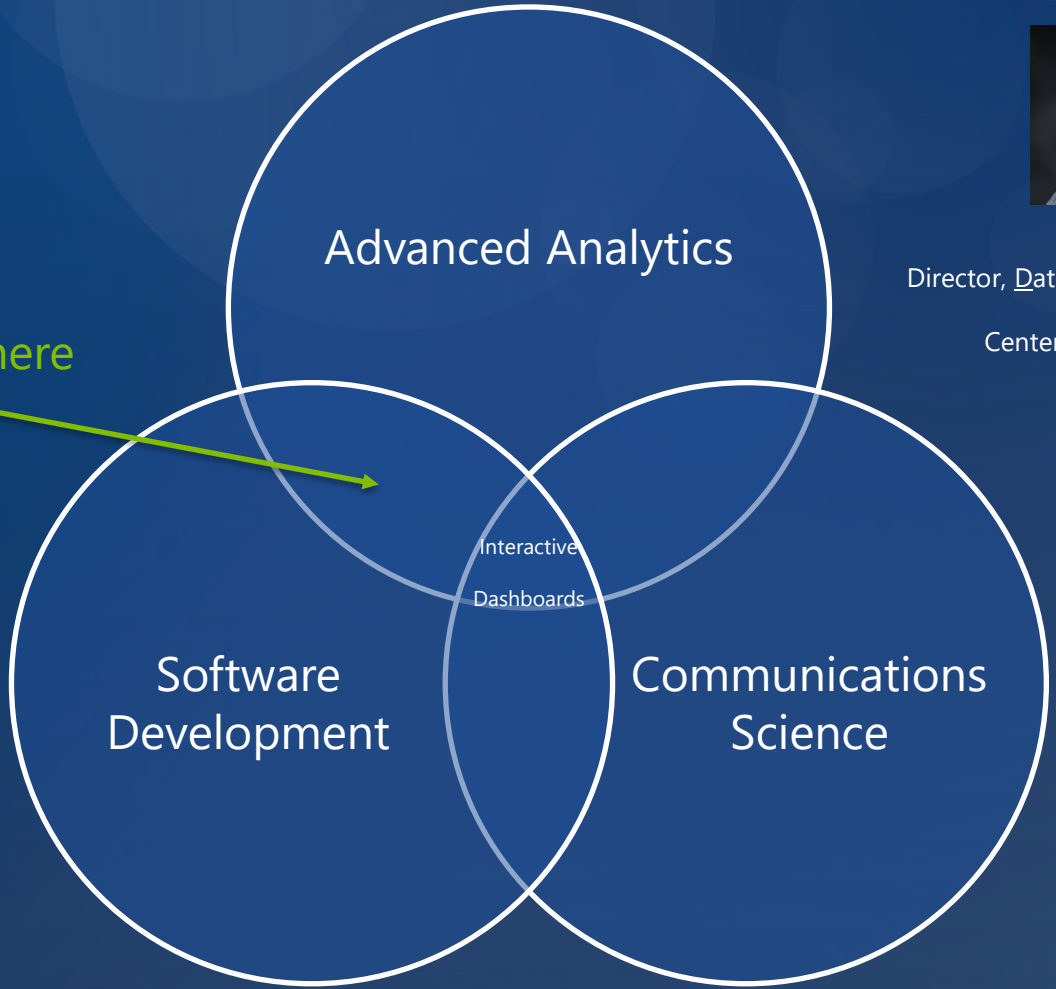

But you might not be here
Advanced Analytics
Or here
Interactive
Dashboards
Software
Development
Communications
Science

Today we're talking about here



Alex Harding
Director, Data Engineering and Visualization Program
Center for Data Science and AI
RTI INTERNATIONAL

I come from here



Data Dissemination Technology

One of the most important factors in the success of a dashboard product is the choice of underlying technology.

BUT…

There is rarely one right answer for which technology to use.

Data Dissemination Technology

Today I'll review:

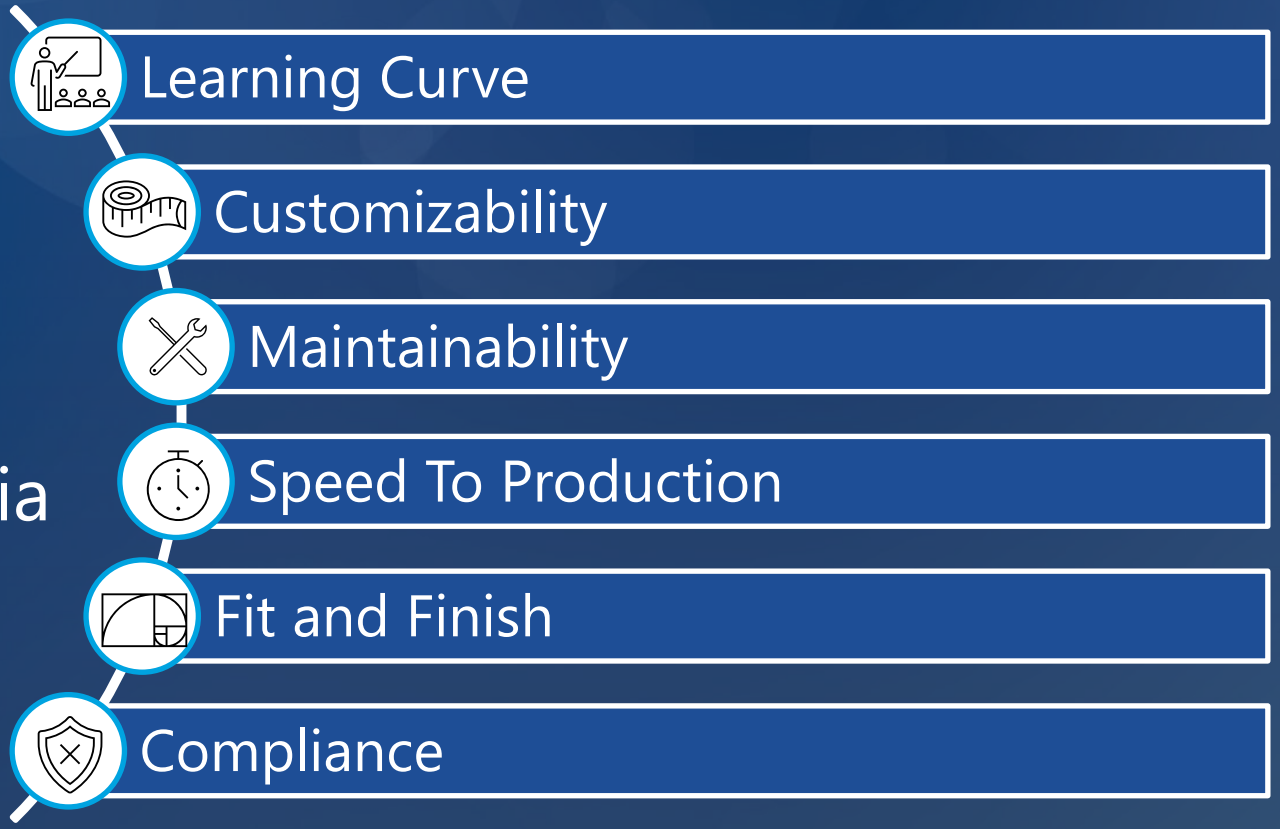
6 criteria for evaluating dashboard technologies

3 categories of dashboard technologies

A comparison of dashboard technologies using the 6 criteria

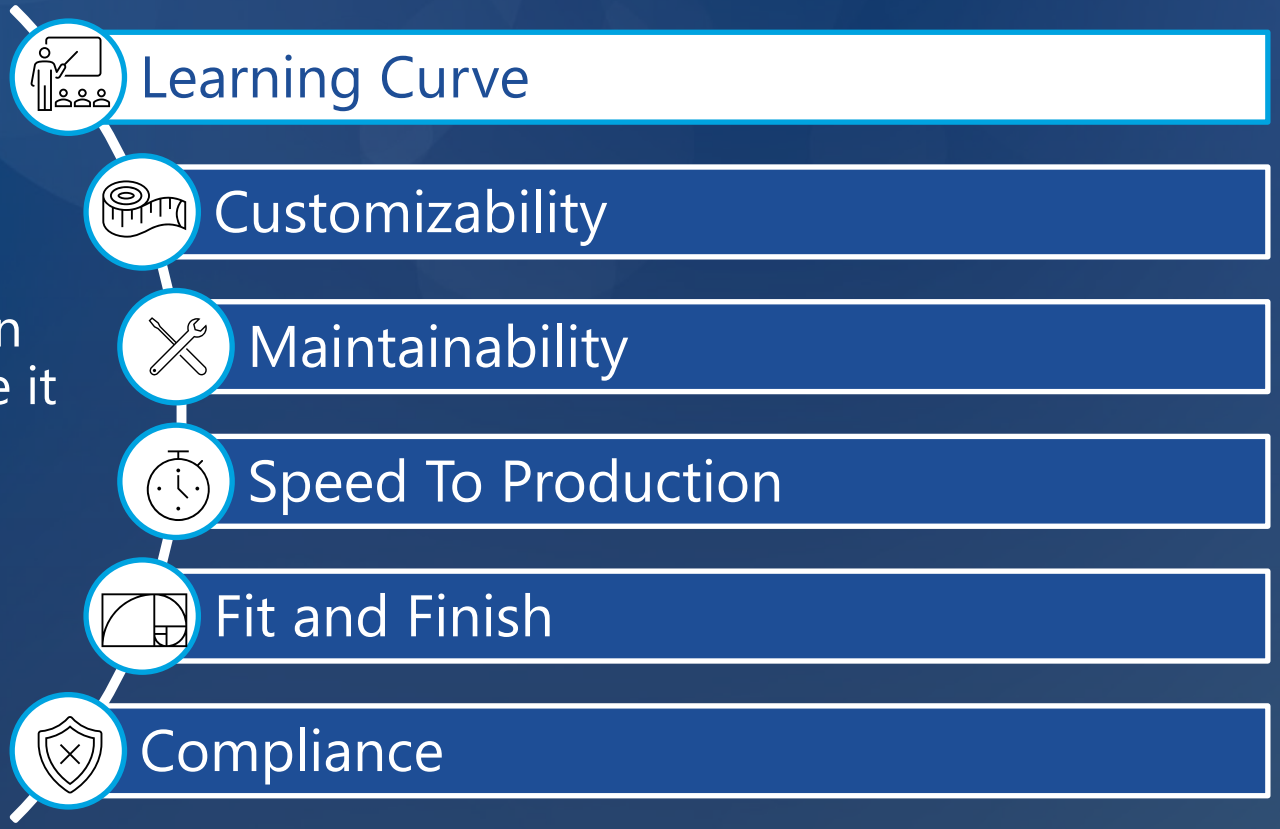
Technology Evaluation Criteria

Dashboard Technology Evaluation Criteria



How easy it is for untrained staff to learn this technology and use it effectively.

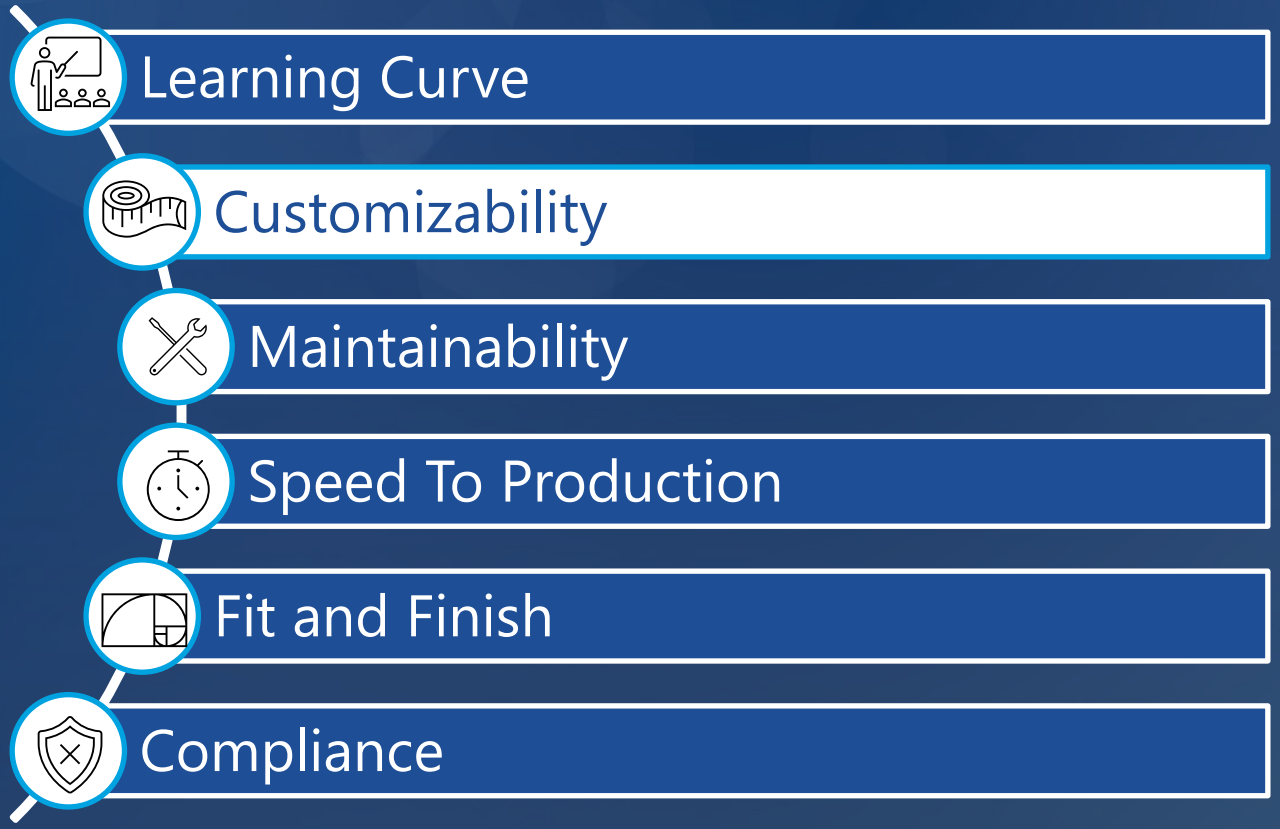
Higher means easier.



How easily customized this technology is.

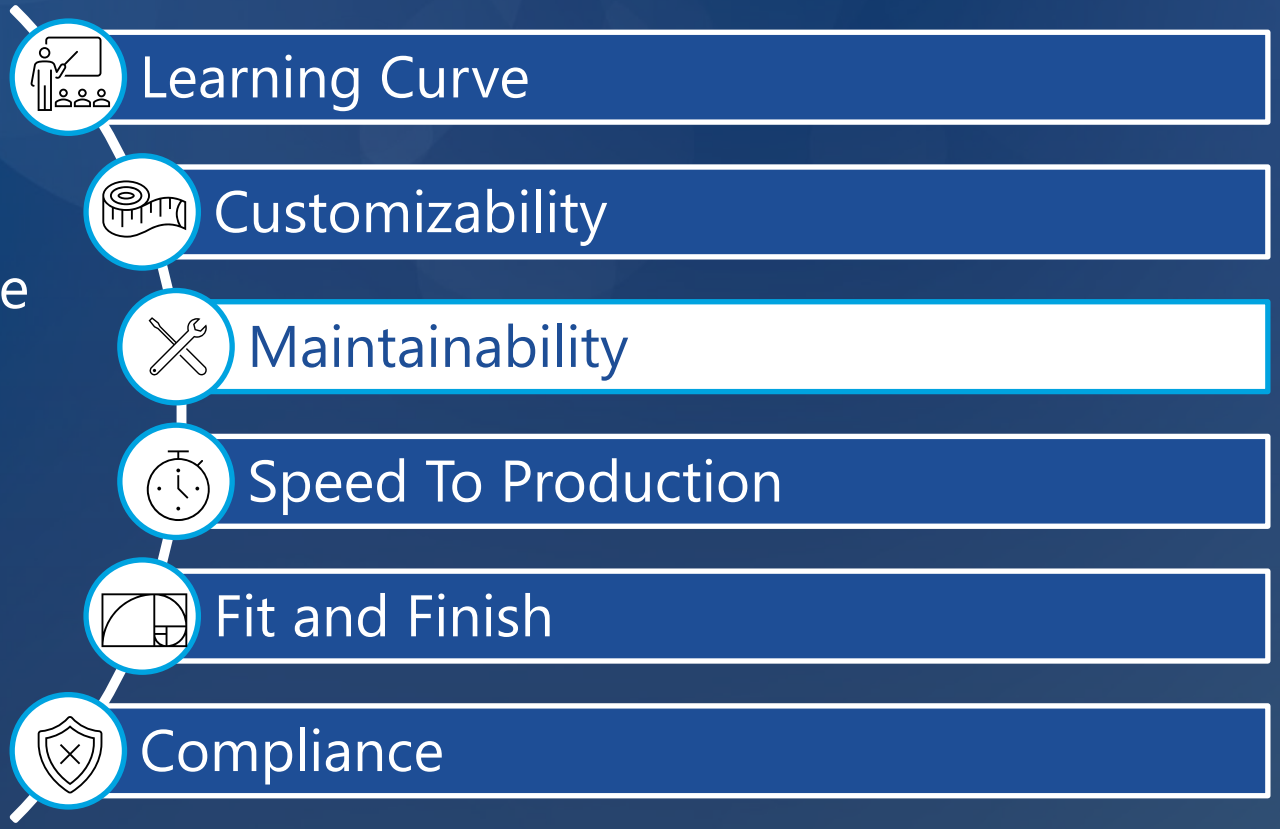
Look and feel, bespoke visualizations, authentication, deployment, scaling, and more.

Higher is more easily customized.



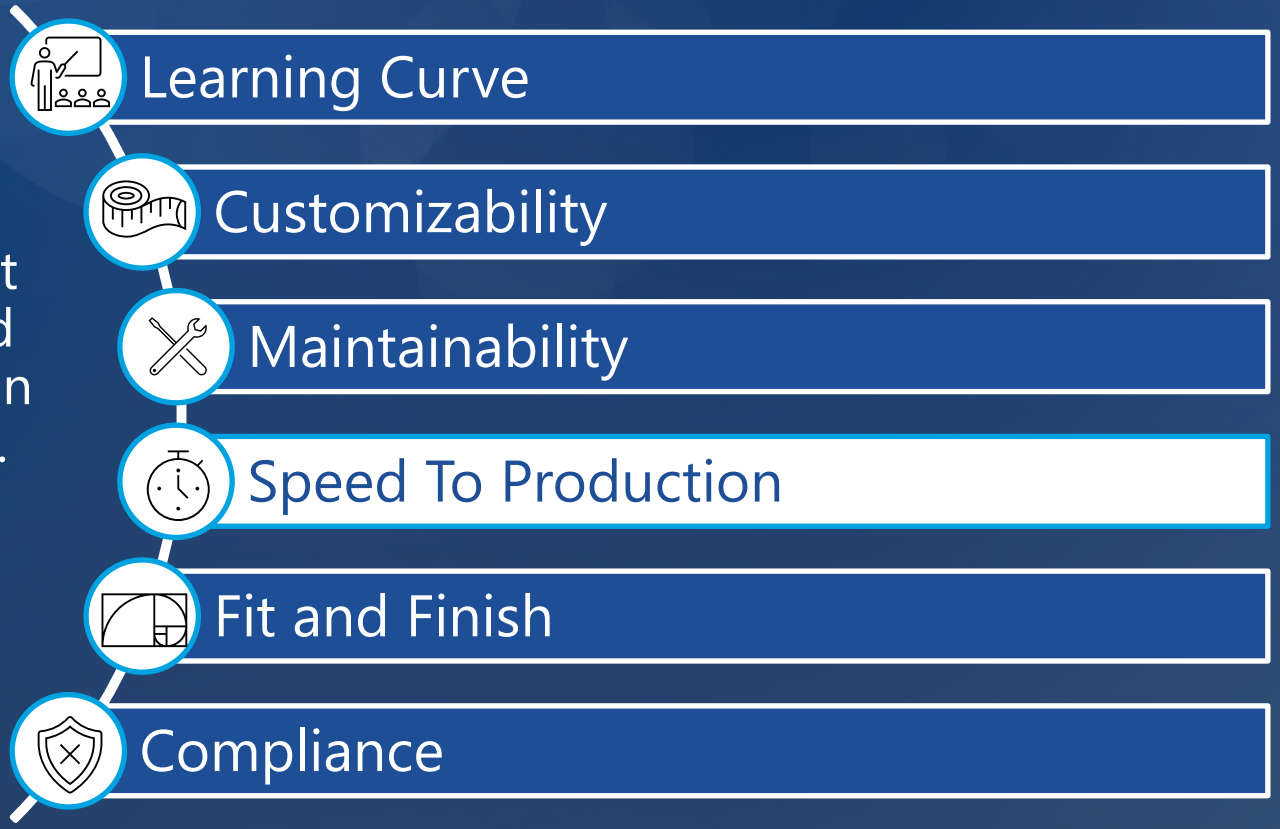
How much labor is required after release to maintain a technology.

Higher is better (less labor).



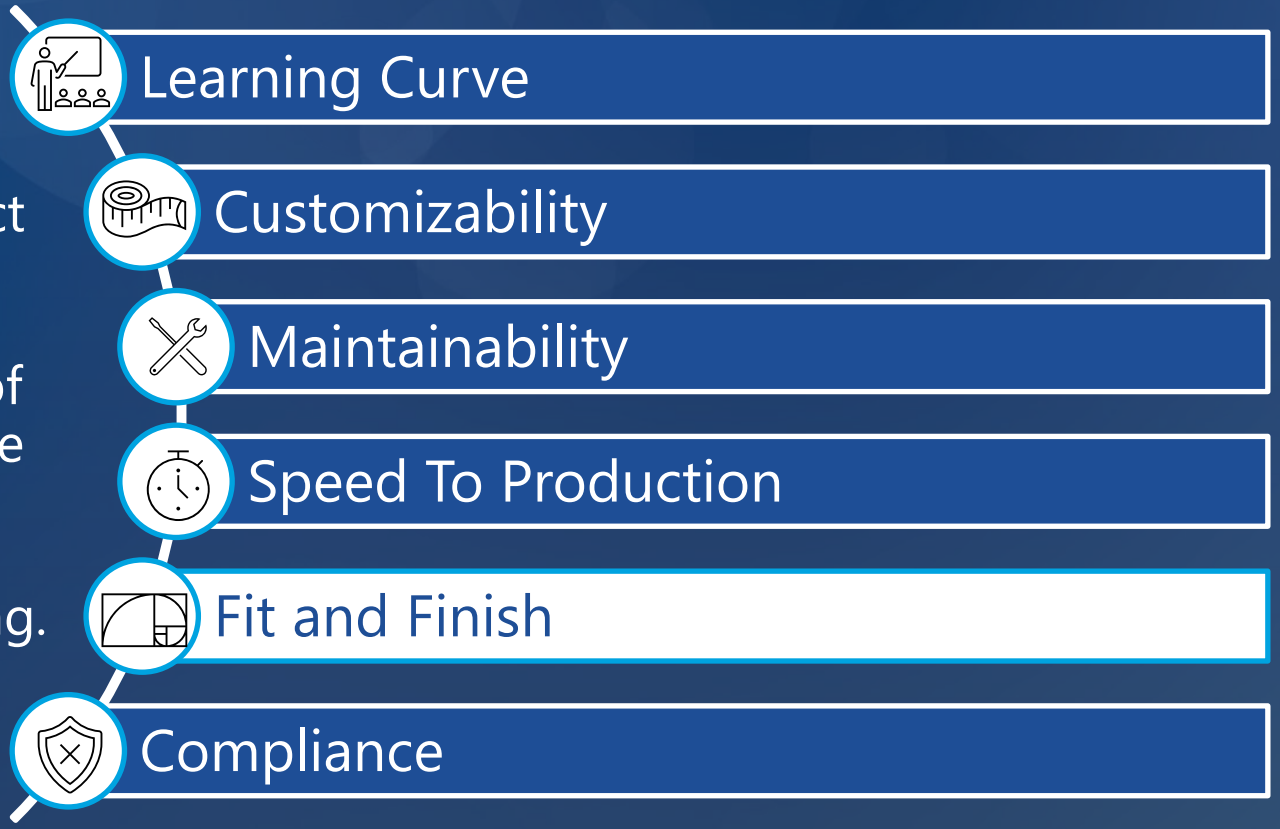
How quickly a product can be developed and deployed to production using the technology.

Higher means faster.



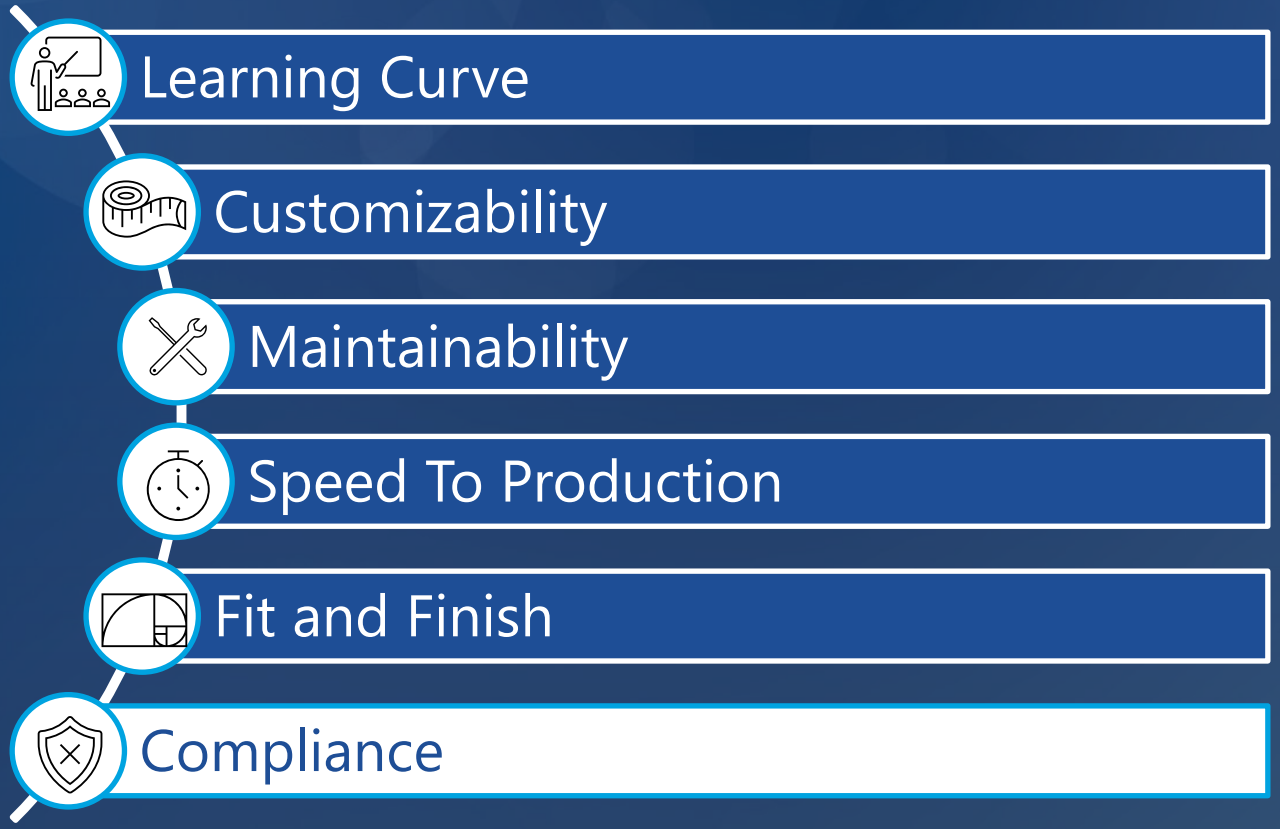
Ceiling for how distinct and attractive this technology can look, given an investment of time and user interface design.

Higher is a better ceiling.



How much control the developers have over factors that determine compliance with policy, such as security and accessibility.

Higher is more easily made compliant.



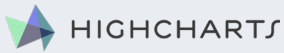
Categorization

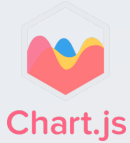
Data Visualization Technology Options

Data Visualization Software (No Code)

Data Science Web Frameworks (Low Code)

Custom Web Dashboards (Custom Code)



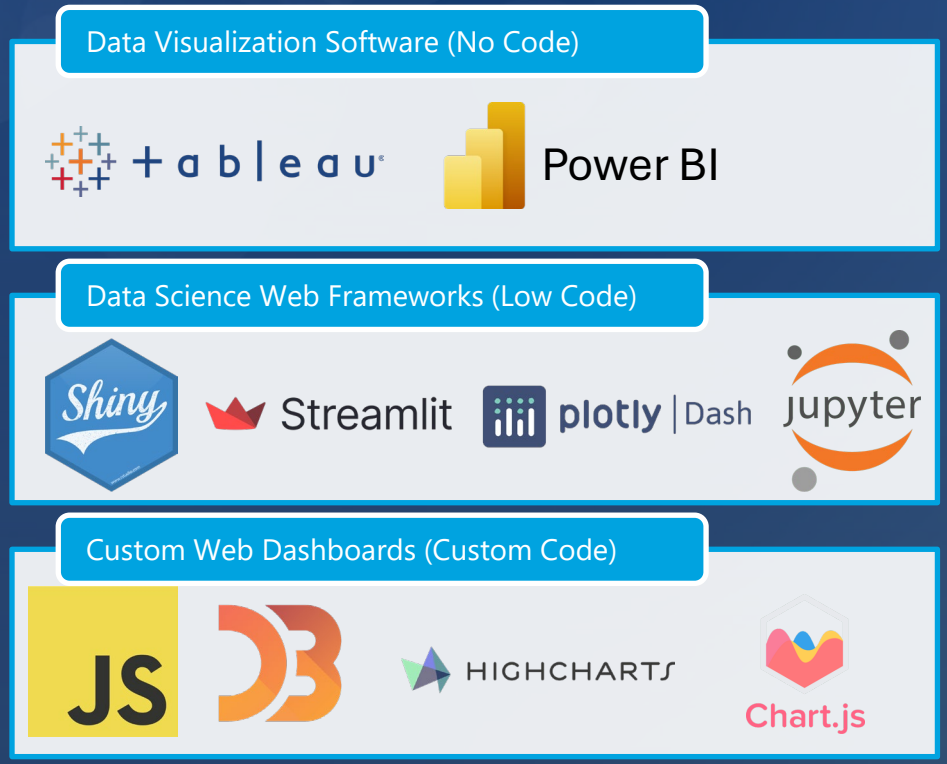


Data Visualization Technology Options

Why these categories?

There are many technologies new, old, and emerging.

By summarizing and generalizing existing technologies, we can create an evergreen set of categories that we can slot technologies into as they emerge.



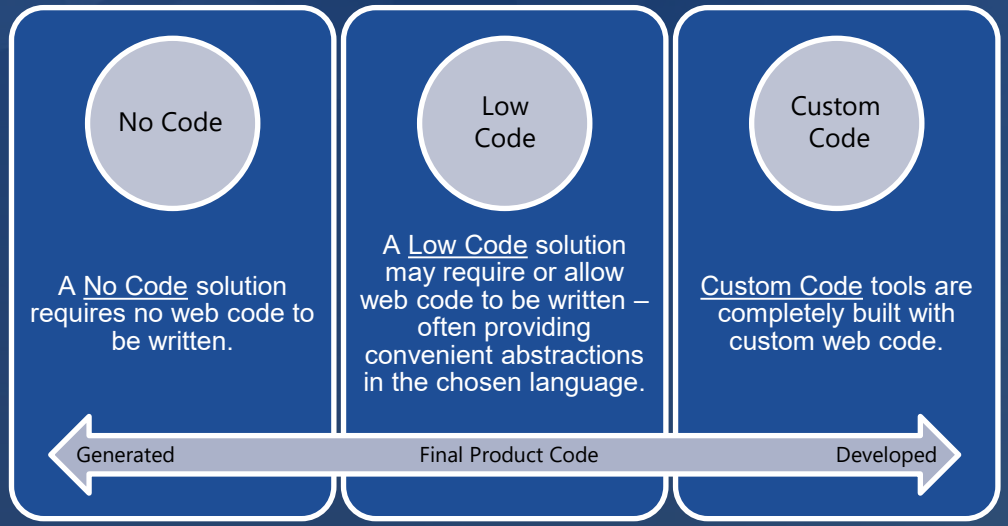
Low/No/Custom Code Designations

Low/No/Custom code refers to the amount of web development code that goes into the tool.

Anything viewable in a web browser is built using HTML, CSS, and JavaScript (client-side web code).

There is a trade-off between how much of that code is generated for you and how much you can customize it.

The low/no/code designations refer to how much of that code is coded by the developer, compared to how much is generated by the tool.



Data Visualization Software (No Code)

Data Visualization Software (No Code)

Data platforms like **Tableau** and **Microsoft Power BI** leverage graphical user interfaces (GUIs) to allow users to create compelling data visualizations without a single line of code.

These platforms often include additional features and add-ons for data analysis, storage, harmonization, and more.

Data Visualization Software (No Code)

Power BI

Pros

- ✓ Easy to use
- ✓ Streamlined staffing mix
- ✓ Low/No long-term maintenance
- ✓ Easy to share and embed
- ✓ Scaling, authentication is built into cloud platforms

Cons

- ❖ Difficult or impossible to customize
- ❖ Limited visualization and storytelling options
- ❖ Data lives on third-party infrastructure or requires significant internal IT effort
- ❖ Data size and performance limitations

Data Science Web Frameworks (Low Code)

Data Science Web Frameworks (Low Code)

Streamlit

plotly | Dash

Data Science Web Frameworks use popular open-source languages such as R and Python to create interactive, web-based dashboards from existing statistical analyses.

These languages, in conjunction with other open-source packages, provide a common framework for both conducting statistical analyses and presenting web-based data dashboards with custom visuals.

Data Science Web Frameworks (Low Code)

Streamlit

Pros

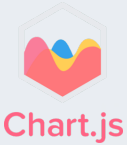
- ✓ Free and open-source technologies
- ✓ Seamless integration with existing statistical analyses and models in R or Python
- ✓ Streamlined and lower-cost staffing mix
- ✓ No web development experience required
- ✓ Extensive library of packages to create data visualizations
- ✓ Rapid prototyping and no separation of prototype and production application

Cons

- ❖ Integration with existing web tools is challenging
- ❖ Customization is difficult and limited
- ❖ Scaling to many users is difficult and/or expensive
- ❖ Rigid language and infrastructure requirements
- ❖ Performance can be difficult to optimize with large data
- ❖ Accessibility and security requirements are dependent on the packages used and are often non-compliant

Custom Web Dashboards (Custom Code)

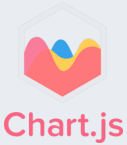
Custom Web Dashboards (Custom Code)



The most comprehensive approach to building a dashboard is a custom web tool built using open-source technologies. When creating a custom web application, there are no limits to what you can and can't do.

This category is heavily dependent on the staff creating the tool, as it has the highest possible quality ceiling, but also the lowest possible quality floor.

Custom Web Dashboards (Custom Code)



Pros

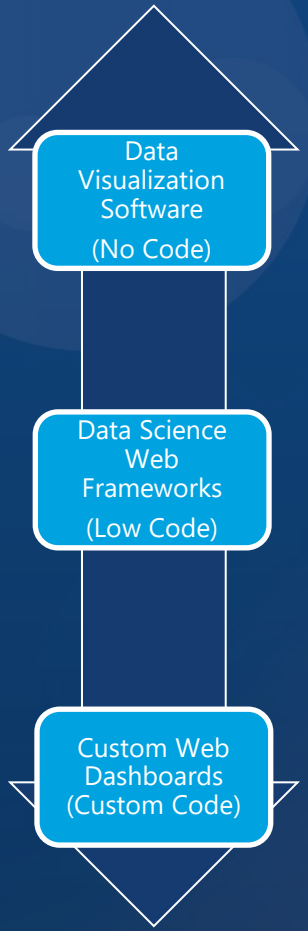
- ✓ Best possible user experience
- ✓ Granular control over all facets of the dashboard
- ✓ Easily integrated with other options or existing web tools
- ✓ Highest possible conformance to security and accessibility requirements

Cons

- ❖ Requires software developers alongside data analysts
- ❖ User experience, fit and finish, design, and performance are all dependent on the developers
- ❖ Longest time to production
- ❖ Highest long-term maintenance costs

Technology Comparison

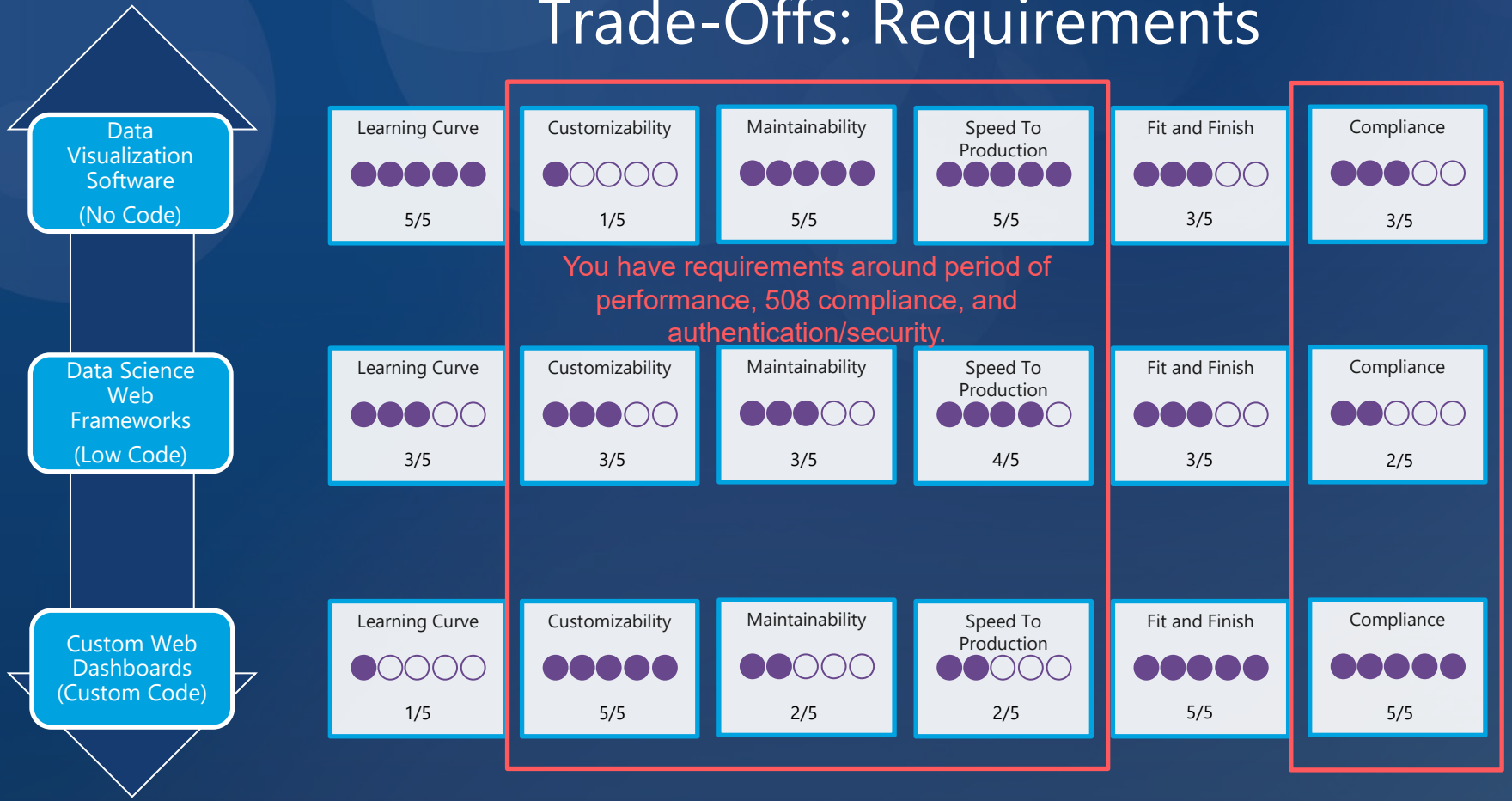
Technology Option Comparison



	Learning Curve	Customizability	Maintainability	Speed To Production	Fit and Finish	Compliance
Data Visualization Software (No Code)	5/5	1/5	5/5	5/5	3/5	3/5
Data Science Web Frameworks (Low Code)	3/5	3/5	3/5	4/5	3/5	2/5
Custom Web Dashboards (Custom Code)	1/5	5/5	2/5	2/5	5/5	5/5

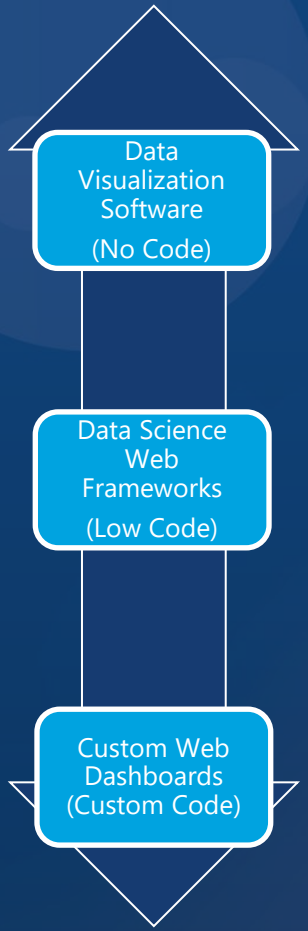
Trade-Offs: Desirability

Trade-Offs: Requirements



Trade-Offs: Capabilities

Use These If…

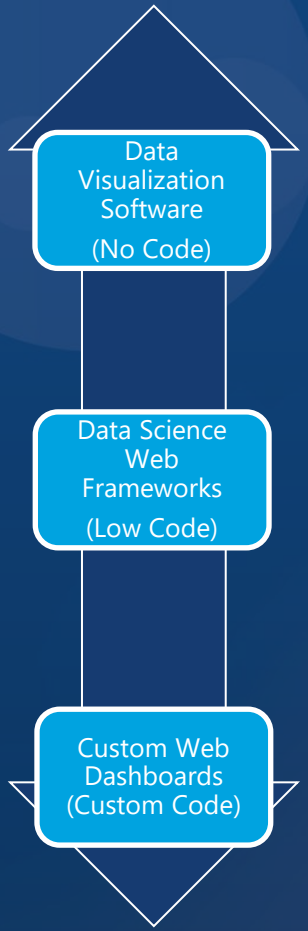


- ✓ Your data stories can be told with the available visualization resources in Tableau/Power BI
- ✓ Your staffing model doesn't support software developers
- ✓ You want a short time from visual analysis to production
- ✓ You want to maintain without contracting developers post-deliverable

- ✓ You have existing analyses in R or Python and staff knowledgeable in these tools
- ✓ You want to share interactive outputs from analyses quickly and easily
- ✓ You want to consolidate cost of analysis and development staff
- ✓ Your staffing model doesn't support software developers

- ✓ You want a highly customized and polished dashboard
- ✓ Your dashboard includes complex interactivity or additional non-dashboard functionalities
- ✓ You need multiple pages with complex navigation
- ✓ You have security and/or accessibility requirements
- ✓ Your dashboard is intended for public consumption or a large user base

Don't Use These If…



- Your data has cloud access restrictions or requires custom authentication
- Your data stories require high visual polish or New York Times style novel visualizations
- Your final deployment destination doesn't support embedding or has rigid deployment criteria
- You have large data and/or complex access requirements like role-based access control

- Your data has cloud access restrictions or requires custom authentication
- Your data stories require high visual polish or New York Times style novel visualizations
- Your final deployment destination requires integration into an existing website or doesn't support R/Python
- You have strict security and/or accessibility requirements (such as 508 compliance)
- You have many users, large data, or complex access requirements
- You have complex interaction or shared data between pages

- You don't have access to software developers
- Your contract requires a short time horizon to production
- You don't have the budget for long-term maintenance
- Long-term maintenance is expected to be handled by non-developers

Takeaways

Takeaways

There is rarely a single right answer for which tool to use, but there is often a wrong answer depending on your requirements, timeline, budget, and capabilities.

Choosing a dashboard technology is a triangulation of your requirements, capabilities, and desires.

First, focus on the dealbreakers to narrow down options. Then, adjust expectations to match the chosen technology.

You can make a good or a bad dashboard in any technology. Choosing the right staff and the right technology for the right staff is crucial.

Thank you

Contact: Alex Harding | email: alexharding@rti.org