

Using software development principles and tools to create an R package for data analysis optimization

Emanuil Markov
Svetomir Hitov
Louise Raiteri

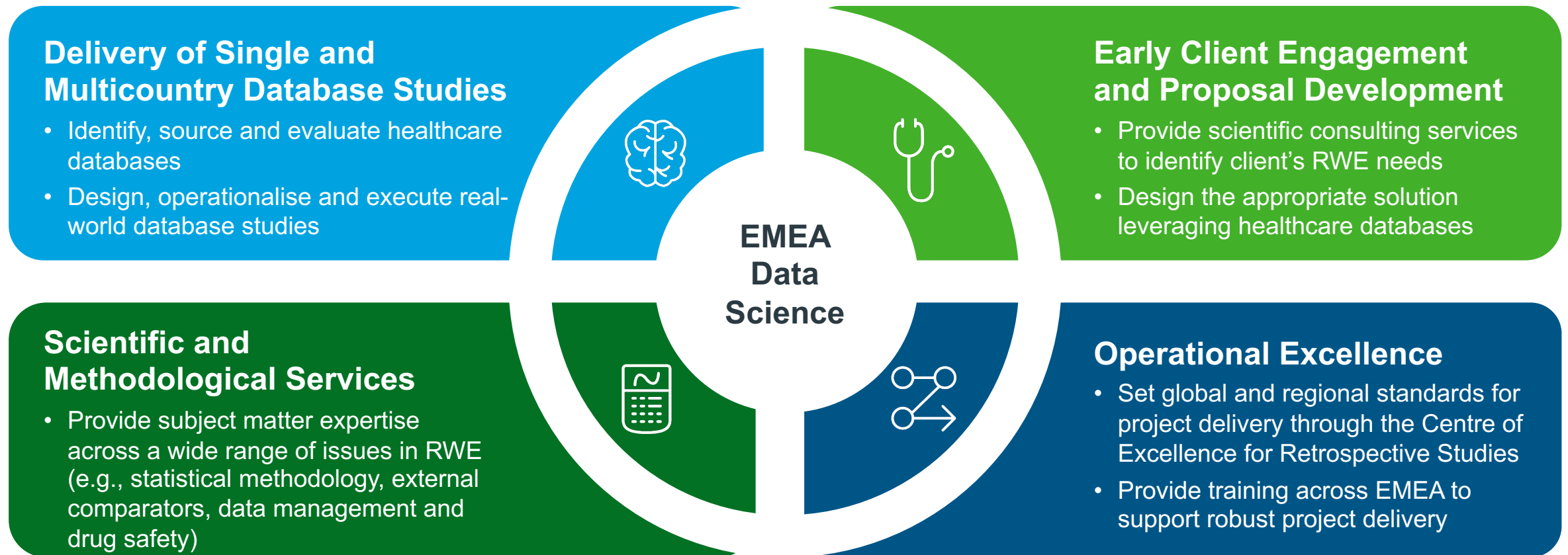


What we do and which principles we apply

- + About us
- + Why we needed to optimize
- + Tools used to achieve optimization
- + Software development principles that fit best to our type of work
- + Outcomes

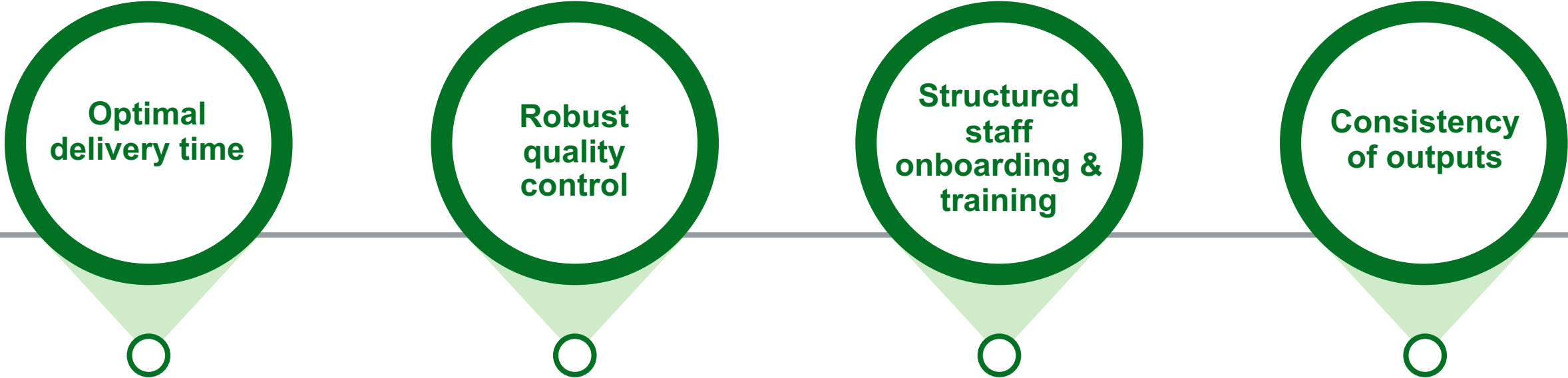
The EMEA Data Science Hub specializes in the design and delivery of scientific studies using real world data

Expertise in working with electronic medical records, existing registries, claims and emerging real world data sources



Data management and statistical programming are key to project delivery and we constantly strive for operational efficiency

How do we code more efficiently across multiple, disparate data sources, to perform the same tasks?



Optimal
delivery time

Robust
quality
control

Structured
staff
onboarding &
training

Consistency
of outputs

We have many tools available to us and many challenges to address

Tools

- R / R Studio / CRAN
- SAS
- Stata
- MS Office suite and Teams
- Jira/Confluence
- Sharepoint
- GitLab:
 - Issue tracker
 - CD/CI
 - Merge requests
 - Project wiki
 - Containerization
- Custom built user interfaces

Challenges

- Varying levels of programming ability across team
- Inconsistent programming styles
- Ad hoc processes for code sharing
- No central repository to store information
- Inconsistent/manual version control eg. _v1, _v2
- Resource intensive training and project handover
- Inefficient code quality control processes
- Inconsistent layout of project outputs
- Lack of time for proper documentation

Introducing **iqviR**: our solution to automating project delivery

*iqviR is an R package with a ***suite of functions*** for reporting*

Descriptive Tables

Analysis and reporting



Table Shells

Standardize creation of shells, with graphical user interface



Small number masking

Preserving patient confidentiality



Sankey Diagrams

Interactive Sankey diagrams to visualize treatment patterns



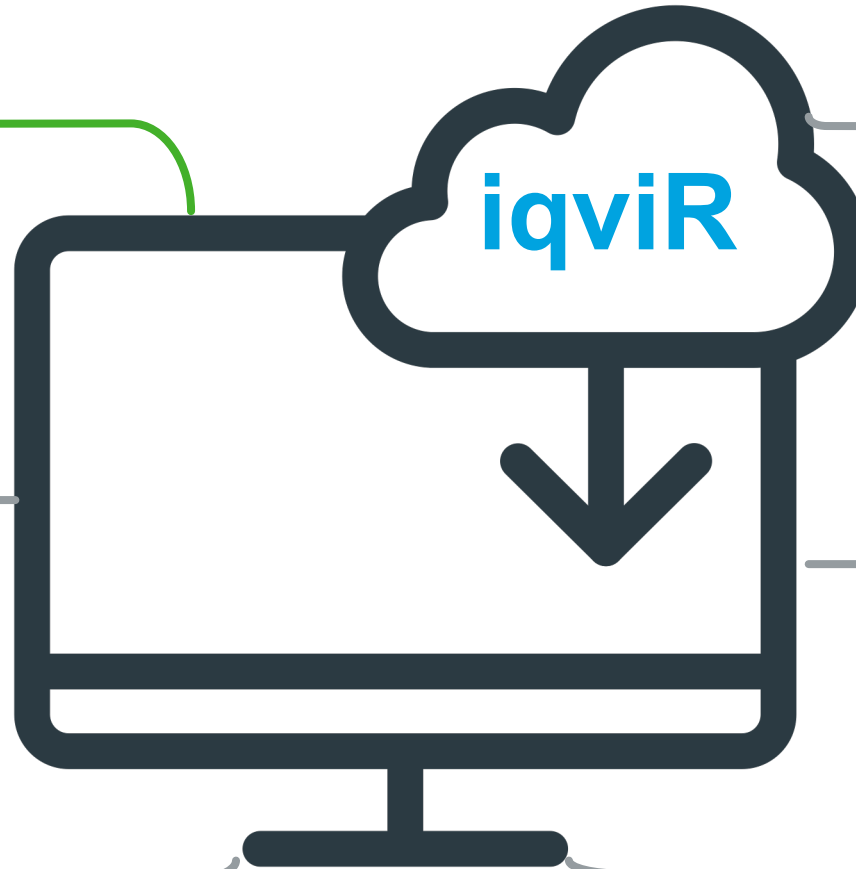
Attrition Diagrams and tables

Visualization with flow diagrams



Kaplan Meier

Consistent high quality KM analysis outputs



The iqviR package

- Generates a standardized reports for the most often used project outputs.
- Provides detailed documentation (help files, vignettes), example data and working out of the box code examples
- Provides similar syntax for the usage of the different functionalities.
- Provides centralized solution for masking and suppression of the results.
- Provides libraries for exporting the results in different formats (Excel, Word, PowerPoint, pdf)

And IQVIR suite of packages

- Data source specific packages - provide templates and reusable code for variable and dataset cleaning, deriving, merging, incorporating the knowledge of the entire team of the dataset.
- User interface tools for:
 - Creation of study diagrams
 - Creation of table shells
 - Creation of attrition table flowcharts
 - Development of project programming specification
 - Code generation for cross-table value checks.

The iqviR development process

- Core team of 6 programmers, application developers and data subject matter experts dedicating approximately 10% of their time to iqviR.
- Code may be developed by ANY programmer in the Data Science team as part of their 'business as usual' data management for projects. Code is then added to the iqviR development workflow for parameterization, function creation and documentation.
- Issue reporting and new feature requests are managed through an issue tracker which is monitored by the core development team
- Any programmer can be involved in development, QC and testing of iqviR

iqviR was built using application development principles

Capabilities to drive improvement



* Humble, Jez, and Gene Kim. Accelerate: the science of lean software and DevOps: building and scaling high performing technology organizations. IT Revolution, 2018.

1

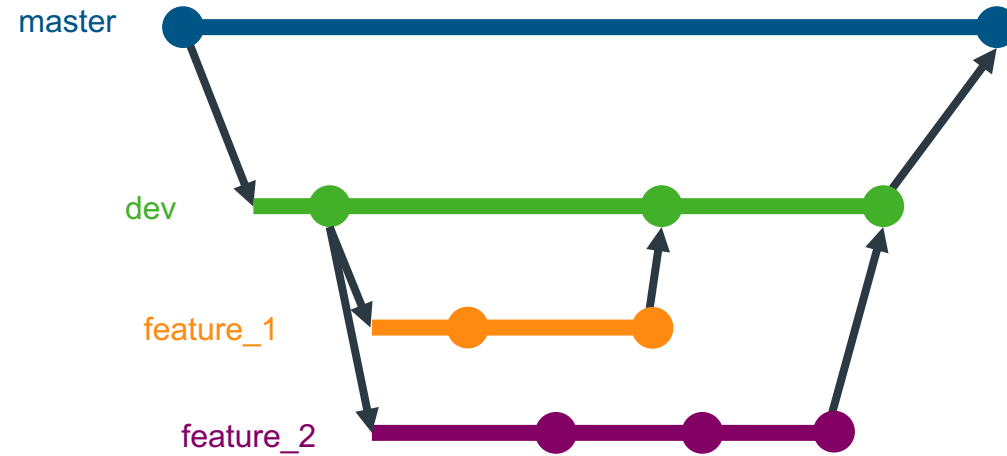
Continuous Delivery

Delivery practices to ensure quality

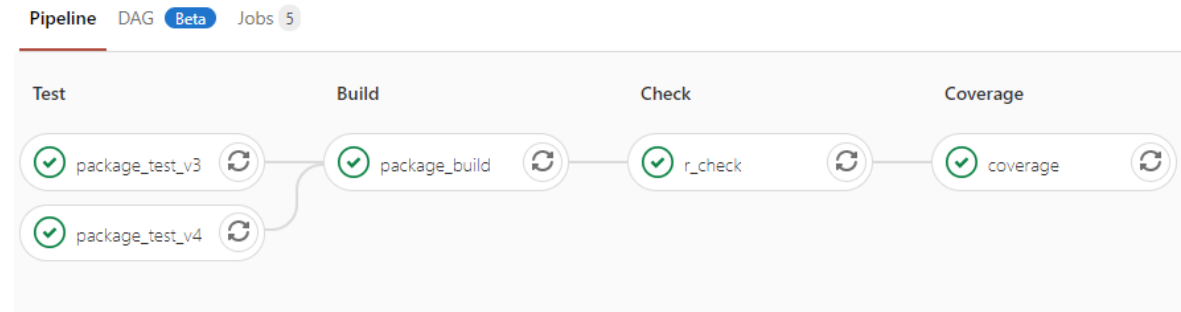
- Version Control
- Test automation
- Test data management

Future considerations

- Trunk based development
- Continuous integration
- Continuous delivery



GitFlow Workflow



2

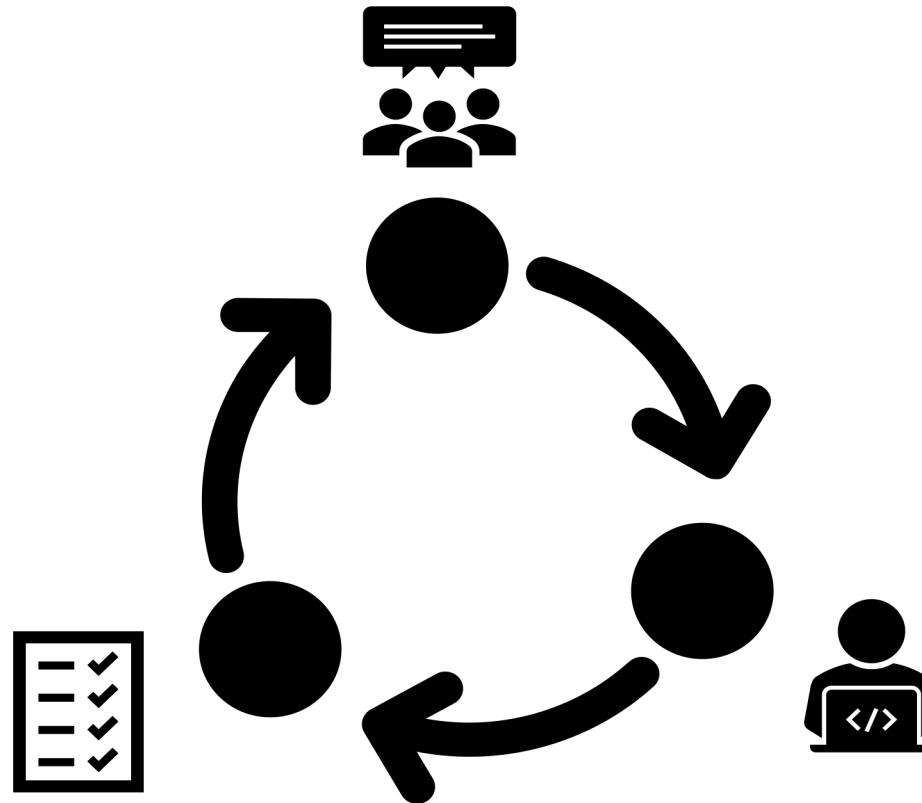
Product and Process

Lean processes to achieve efficiency and satisfaction

- Foster team experimentation
- Customer feedback

Future considerations

- Value Stream Visibility
- Work in small batches



3

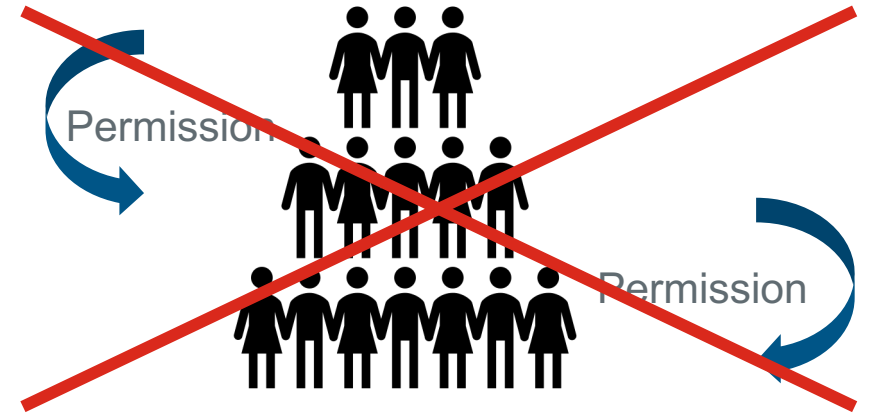
Management and Monitoring

Flexibility between management and execution

- Lightweight change approval
- Visualize work to monitor quality and communicate throughout the team

Future considerations

- Monitor application to inform business decisions
- Manage work with work-in-process limits



4

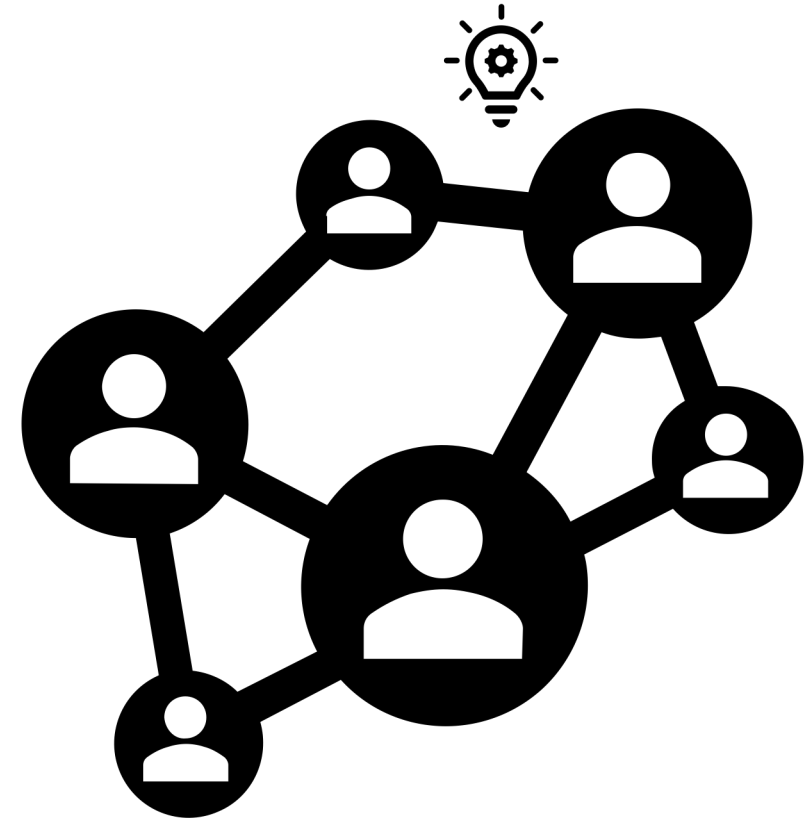
Culture

Friendly and supportive culture

- **Generative culture**
- **Encourage and support learning**
- **Facilitate collaboration among teams**

Future considerations

- Provide resources and tools that make work meaningful
- Support or embody transformational leadership



5

Architecture

iqviR is a stand alone application developed and deployed without dependency on the IT department

Not Applicable

- Loosely coupled architecture
- Empowering teams

What are the benefits of *iqviR*?

Programmers can focus on the important stuff

Reporting is *standardised*, so no need to worry about formatting. Reduced programming time and easy to work with SAS datasets



Easy to install and distribute

Downloaded from a central controlled repository and loaded with 1 line of code

Facilitates QC of code

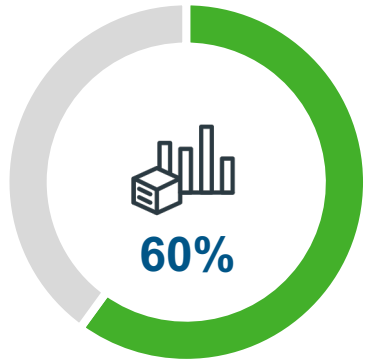
Programmers review much fewer lines of code (10's rather than 100's or 1000's); outputs are predictable making clinical/epi QC easier



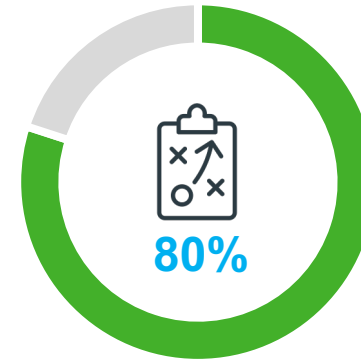
More robust deliverables

Programmers can deliver consistent, high quality deliverables regardless of geographical location or experience

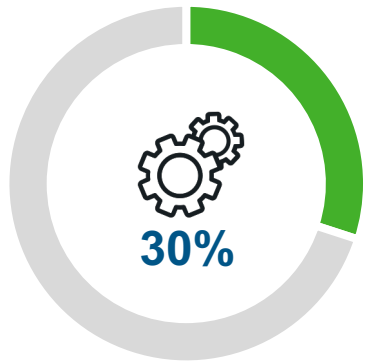
Productivity gains are significant



60% reduction in time to create and print standard descriptive tables compared to non-iqviR solutions by re-using table generation code



80% reduction in time to create table shells, by ensuring a consistent look, simple updates, and identical appearance to final deliverables



30% reduction in time to create bespoke table deliverables by building off standard templates

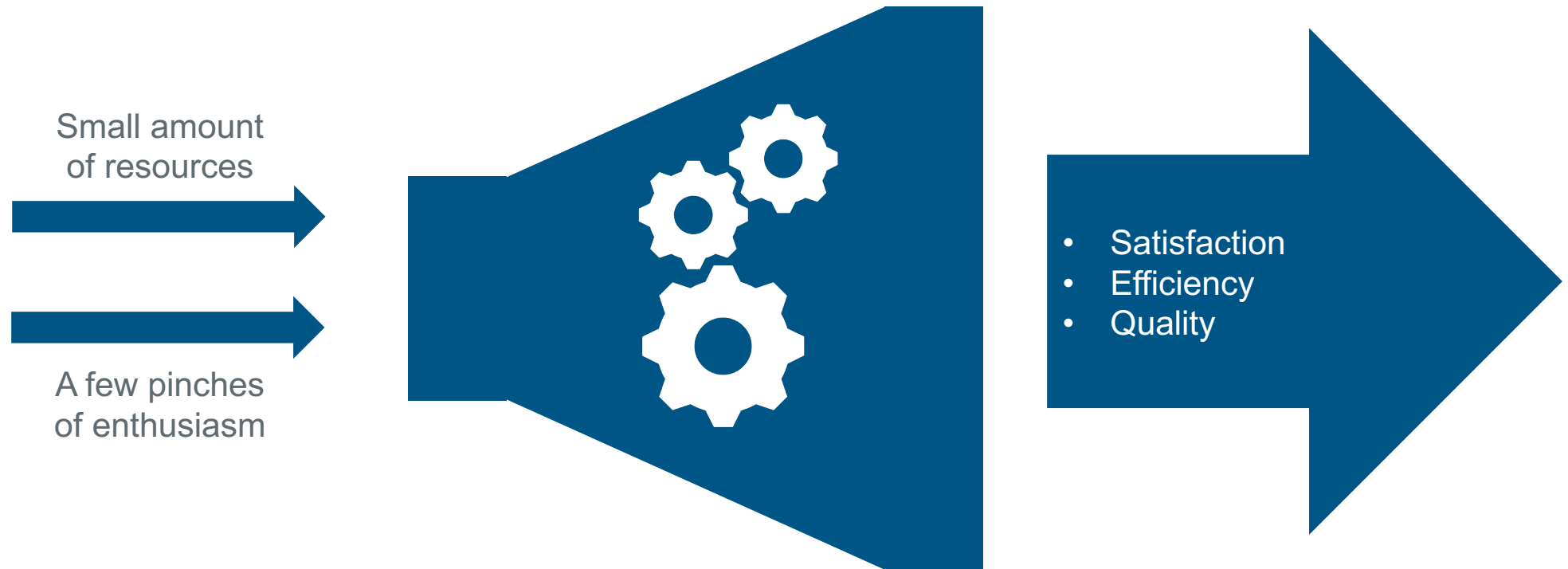


10 – 70% reduction in QC time by elimination of simple errors and increased consistency and quality

Fully auditable QC process using GitLab

Standard coding structure enabling easier project handover

Outcomes





Thank You

Emanuil.Markov@iqvia.com
Svetomir.Hitov@iqvia.com
Louise.Raiteri@iqvia.com