PARTS BEYOND CODE:

Crafting Sensible Statistical-Scientist-Led Automation with Shiny in Pharma

X. Gregory Chen [1], Qian Wang [2], Julian Schmocker [1], Adrian C. Lo [1]

[1] MSD Switzerland, [2] MSD Belgium

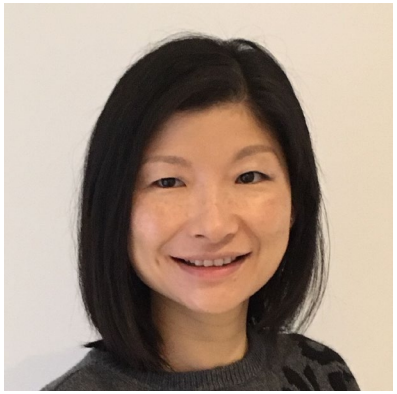
DECLARATION

This presentation only reflects our personal view or understanding, not representing company's position.

MEET THE AUTHOR TEAM

Gregory Chen

Principal Scientist
BARDS HTA Statistics



Qian Wang

Senior Principal Scientist
BARDS Stat. Programming

Julian Schmocker

Senior Scientist
BARDS Stat. Programming

Adrian Lo

Scientist
BARDS Stat. Programming

ACKNOWLEDGEMENT

The App mentioned here is developed by a group of BARDS HTA Statisticians and statistical programmers at MSD.

AGENDA

Demo of an Example

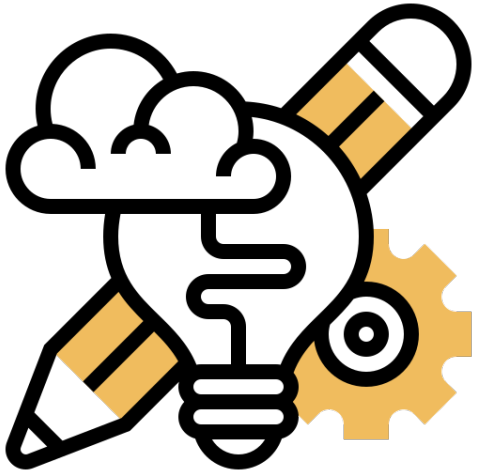
Product Design

Development Collaboration

Change Management

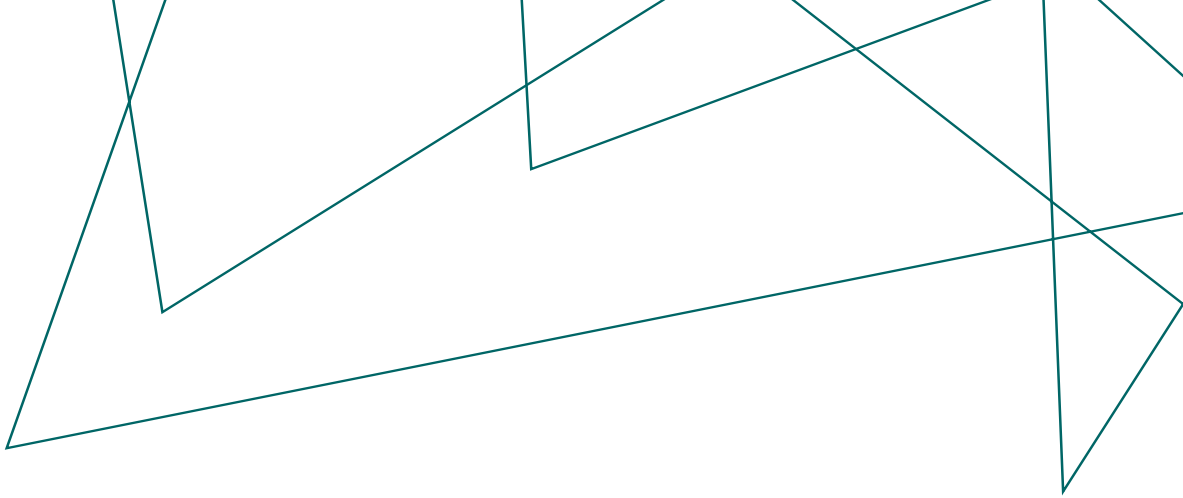
UX, Validation, Feedback

DEMO OF AN EXAMPLE H2H PORTAL



PRODUCT DESIGN

CLEAR PURPOSE OF AUTOMATION



What we knew

- Automation **cannot fully replace** human effort in an A&R process
- Over-ambitious and poorly-designed automation does more **harm** than **benefit**

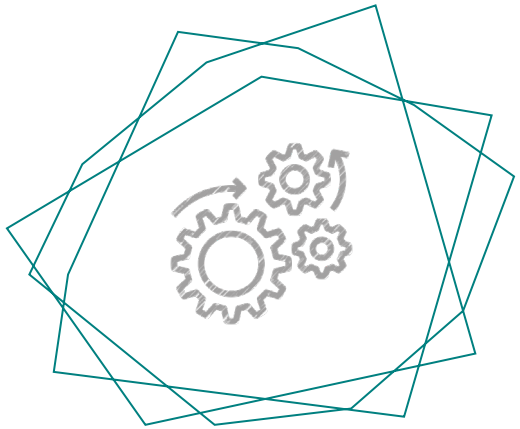
What we want to achieve

- **Fit for purpose** and lightweight app that leverage Shiny framework in R that would not require big excursion from a typical curriculum of a statistical scientist. General software is more an IT project
- **Free up** statistical resources from "**Routine**" and "**Repetitive**" work. Enable resources to **focus on strategic** (analysis planning and interpretation) and **complex** (non-standard analyses) tasks

EXAMPLE APP: H2H PORTAL

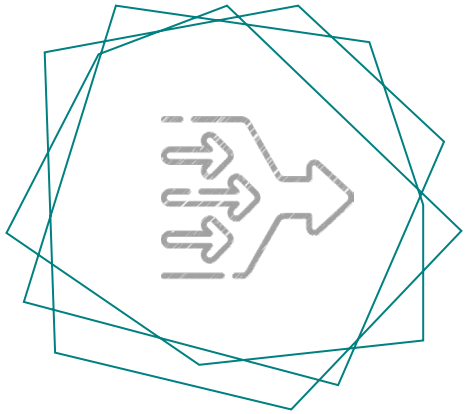
- **an attempt to**
  - ✓ automate most of the operational steps in an A&R life cycle
  - ✓ integrate scattered templates, as well as SAS and R execution
  - ✓ build up a master model of A&R metadata to interlink study assets (code, documents, data, operation), for easier impact assessment and cross-project inquiry
- **an experiment to** (know what and how by trying)
  - ✓ what steps in A&R make sense to be automated
  - ✓ to what degrees we should automate to really achieve efficiency and productivity gain, given a target group of users in mind
  - ✓ of what level of complexity the tool still make sense to be built by our statistical function

KEY USER STORIES



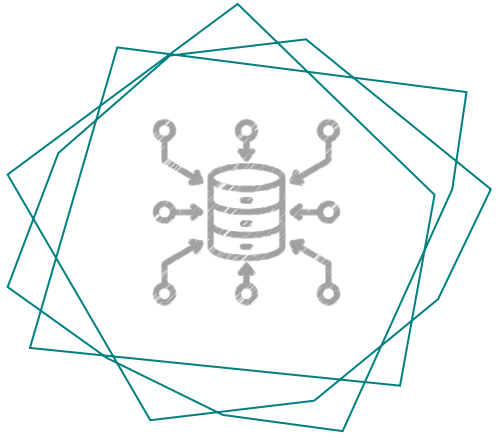
AUTOMATE

Routine and repetitive tasks
With Transparency and Reproducibility



SINGLE ENTRY

With Guidance and Check



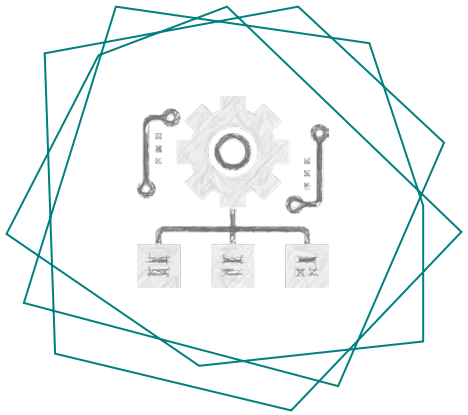
CENTRALIZE METADATA

With Versioning

KEY DESIGN

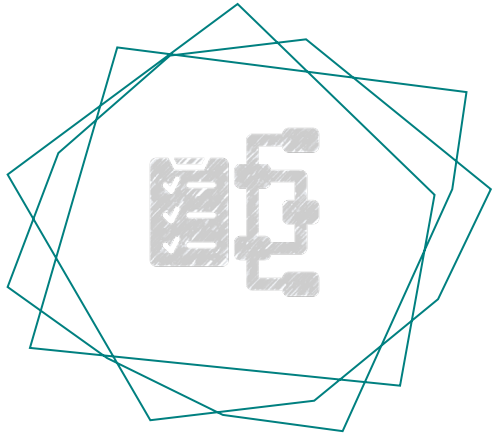
MODULARIZATION

For Scalarability and Maintenance



STRUCTURED METADATA

For Reproducibility and Knowledge

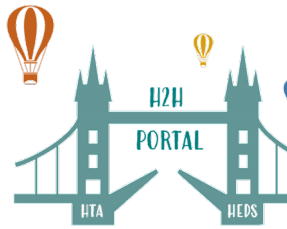


DEPENDENCY MGMT

For Agility of Updates

MODULARIZATION

App Suite for one line of work



H2H Portal App + {mkhecon} R pkg

HECON descriptive analysis

H2H-SEA App + {mksurvce} R pkg

Survival Extrapolation

Filemover App

HECON result delivery

Each App

Standalone

R functions (macros)
Functionality that need to be used repetitively, or something to be **maintained individually**, e.g. what desired format of a process doc the metadata should be exported

Shiny Modules
A mini app. If appropriate and designed well, a complex app can be built up mostly by assembling several Shiny modules.

Database of Standards
Database of standards and external information are NEVER hard coded, but store in **separate non-R files** to be sourced in the app, with R scripts to accelerate future update

App-specific Code
Code that are specific to H2H that cannot be directly used in another app

App-specific Assets
Logo, pictures, snippets of explanatory text (saved as .md file) to make the interface informative and look nice

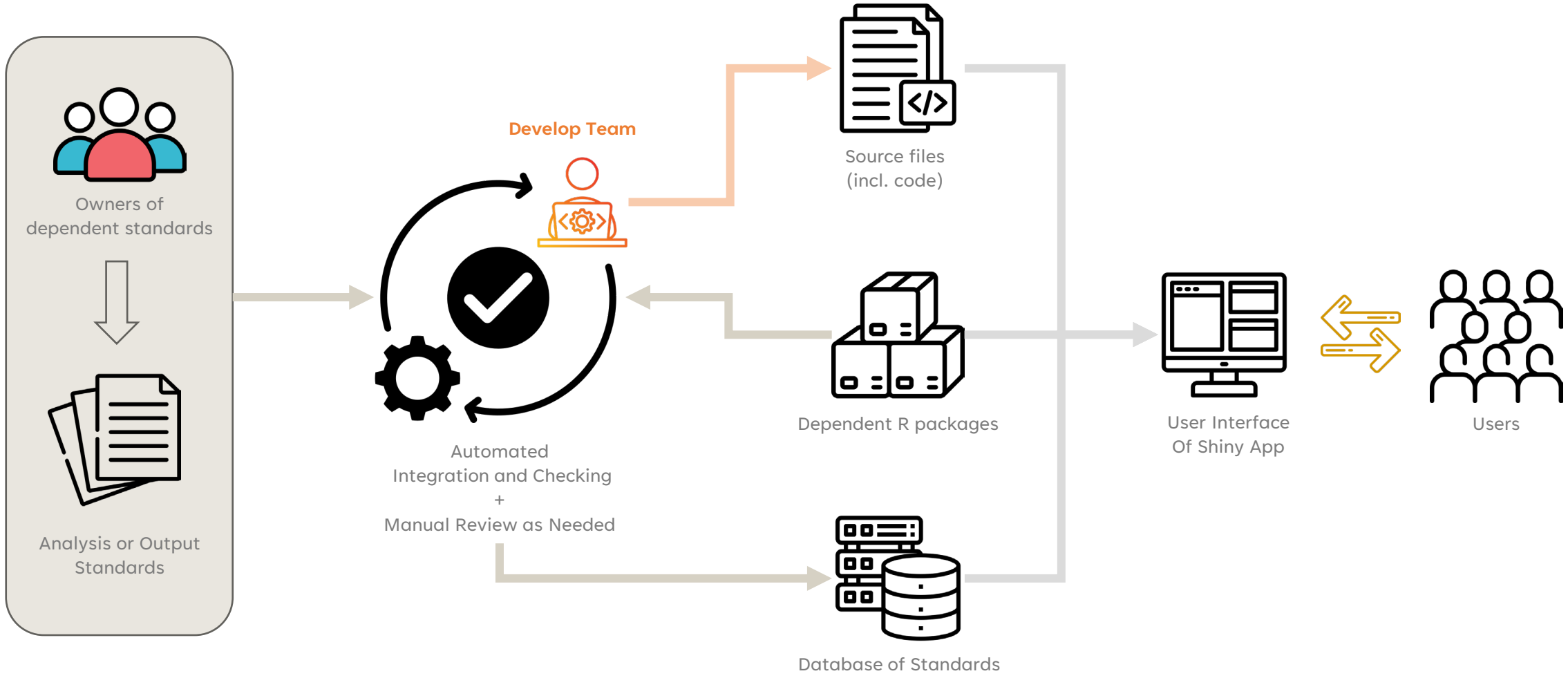
STRUCTURED METADATA

Metadata, in the context of transparent and reproducible automation of data analysis, refers information that documents various aspects of the data and the analysis processes used within the app. This includes descriptive, structural, and administrative details that help users understand, verify, and reproduce the analysis.

Setup in the example app is as listed on the right.

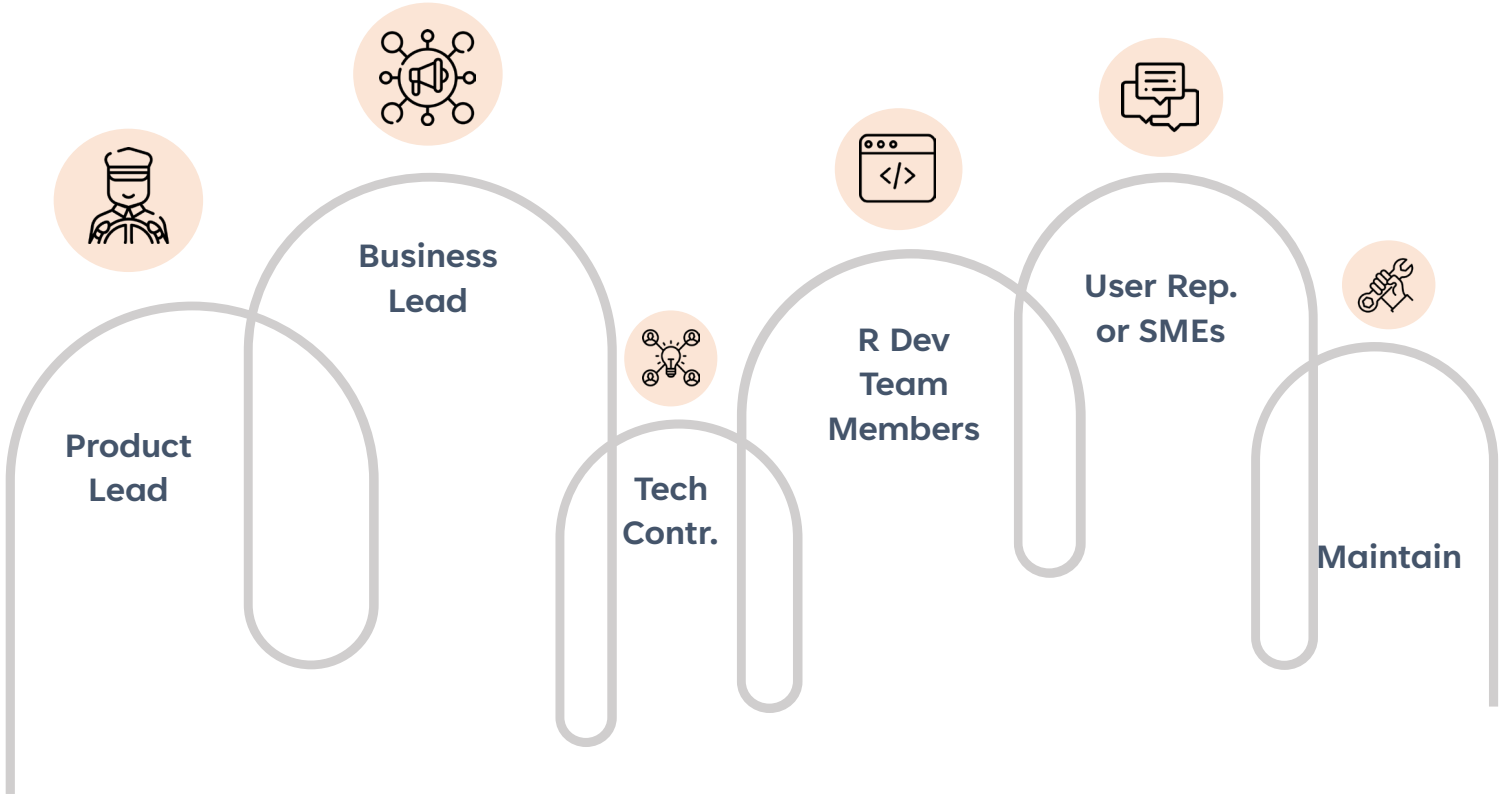
	Key Content
Data Meta	▪ Data source and content information, including specification of how a new variable should be derived based on raw
Analysis Meta	▪ Analysis plan, incl. Population, Intervention, Comparator, Outcome (PICO) definition ▪ Methods and modeling specification ▪ Template output (e.g. mockup table and figure) ▪ Programmatic implementation (e.g. software, dependencies, programmatic setup, version)
Output Meta	▪ Title, subtitle, footnote ▪ Column formats (e.g. decimal places, missing value convention, truncation) ▪ Output file name, location, time stamp ▪ Position in a report
Operation Meta	▪ Contextual information of the study ▪ Who work on what, when ▪ Timeline for deliverables ▪ Status of deliverables

DEPENDENCY MANAGEMENT



DEVELOPMENT COLLABORATION

STRUCTURED PRODUCT TEAM



CLEAR ROLES AND RESPONSIBILITIES

Example in our product team

Proprietary

New Roles & Responsibilities

Product Lead

- Tech lead, typically who initiate or co-initiate the idea, elaborate the design, capable to make prototype, and assess feabilities and impact of needed spec/features of the product
- Get business buy-in together with Business Lead
- R Project Lead in the Dev team, responsible for and oversee the entire SDLC of the R product, its branding and marketing, and its version plan, together with Taskforce Leads.

Dev Team

- Techincal team to support Product lead throught its SDLC
- Depend on team size, may incl. a ST/SP lead, who would do the mapping of functions and give skeleton of spec of each function, critical review the tech design of the product
- Each member independently deliver at least 1 functionality or feature, or majority of unit tests
- Knowledgeable about the product

Tech Contributor

- Tech consultant, Pull request review, Quality reviewer, help with testing

Business Lead (for App)

- (Co-)Initiate the idea and design with Product lead to facilitate in getting business buy-in
- Coordinate other key business stakeholder/SMEs
- Share the responsibility of branding and marketing of the tool, maximize its impact to business
- Participate in version planning

SMEs

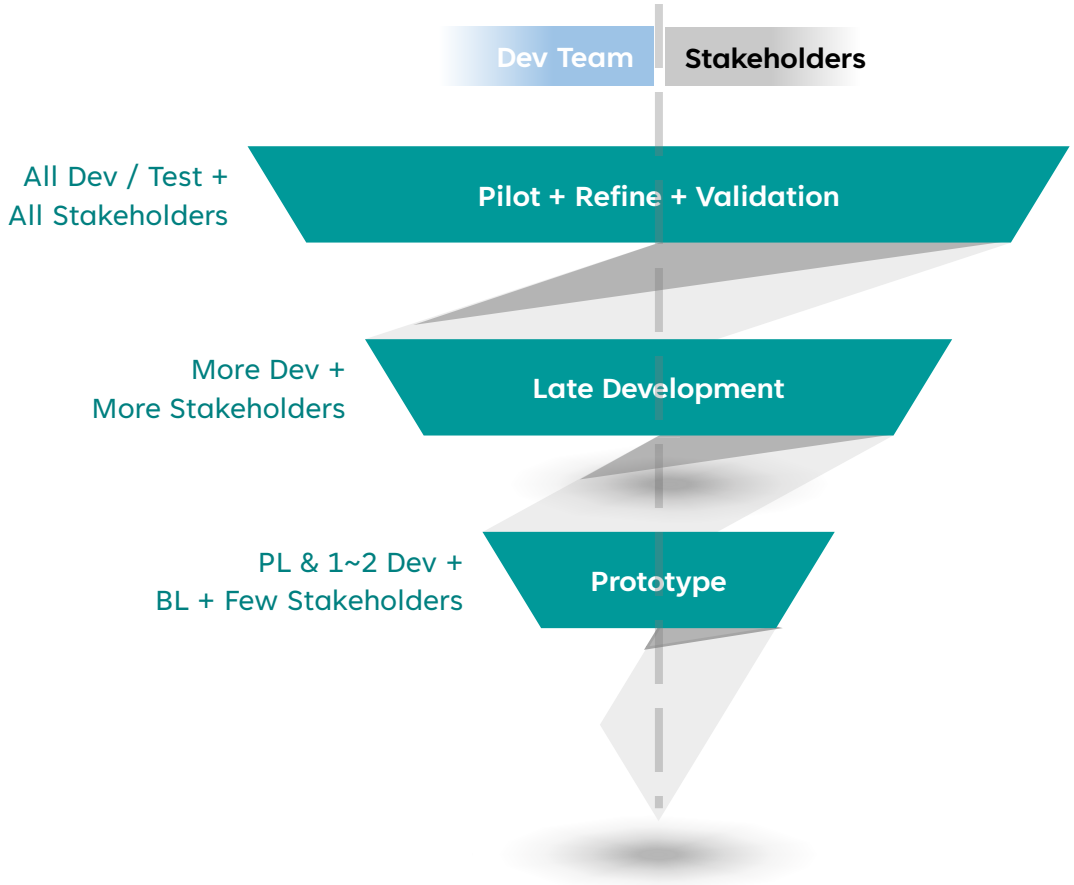
- Related initative leads or other business key stakeholders.
- Methodological SMEs, or project leads that are crucial to pilot the R product

Maintainer

- Key contact to customers for production versions. With tech capacity to answer tech questions, resolve tech issues or coordinate with Dev team to resolve, collect bug report and feedback to Product lead

SENSIBLE ENGAGEMENT IN DIFFERENT PHASES

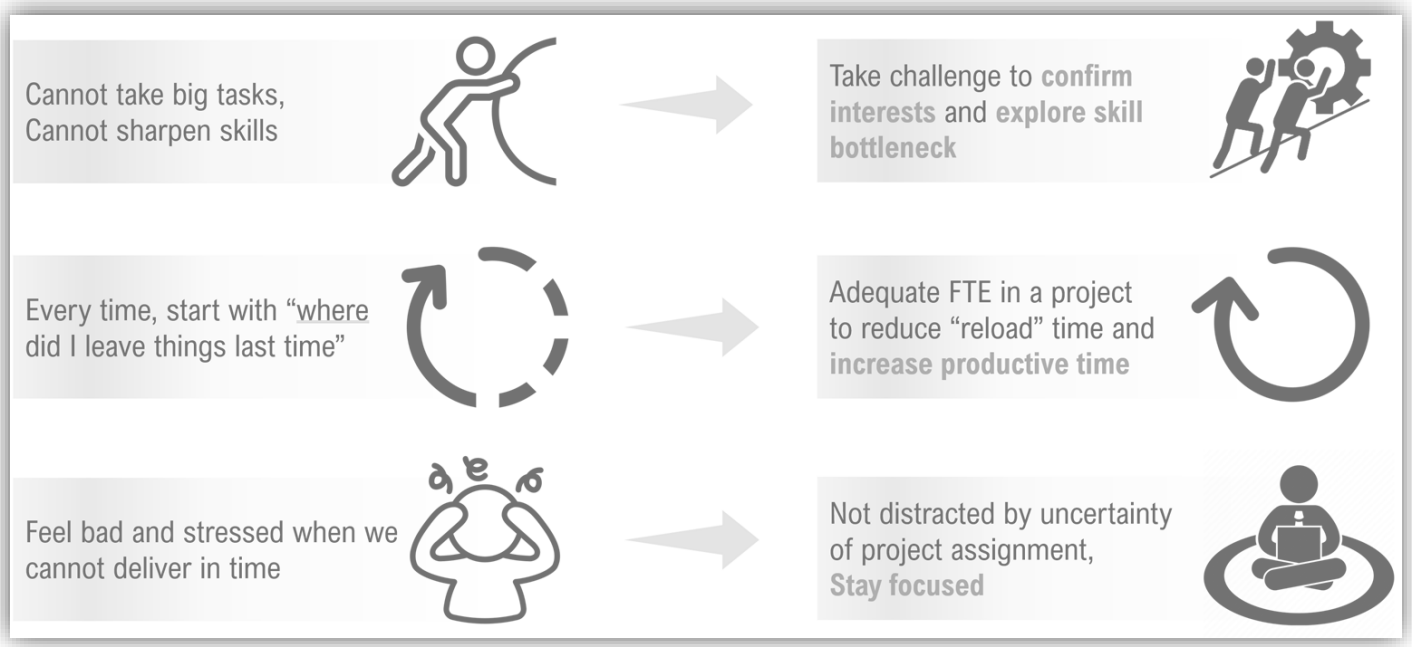
"MORE PEOPLE" ≠ "FASTER"



EFFECTIVE RESOURCING MODEL

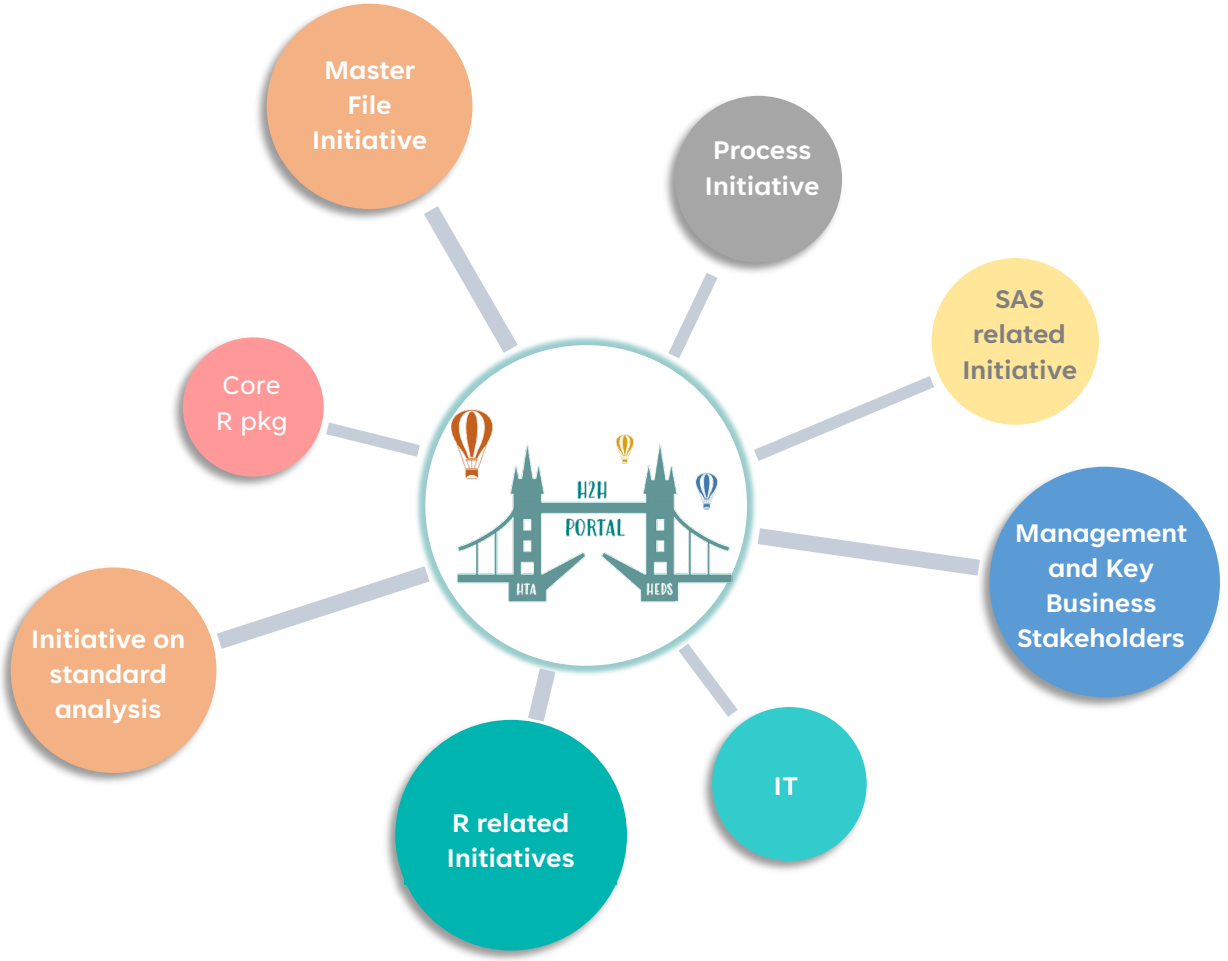
Right talents, with right skill set, for the right role

"Large team and each member has a small percentage of dedicated time" does NOT seem to work well



CHANGE MANAGEMENT

EXAMPLE:
A COMPLEX APP
WITH MANY DEPENDENCY



CHANGE MANAGEMENT

- **Early engagement of key stakeholders**
- **Prototyping**
- **Playground**
- **Piloting**
- **Training**
- **Early adopter support**

UX, VALIDATION, FEEDBACK

What to Validate for an App

a Shiny App ≈ **a R pkg** (a set of R functions) + **User Interface** (a webpage alike frontend)

Functionality

Back-end features of a Shiny app. For example,

- make prescribed simulation, calculation, modeling, plotting
- reporting related functions (generate a table or file that can be used in an official document)

Functional test

Test the specified functionality features, e.g. correctness of the calculation, modeling outcome, plot and text.

User Features

Features of the UI that users/customers can interact with and achieve certain outcome (e.g. switch plot from A to B, filter table, download a report after clicking a button).

User Acceptance Test (UAT)

In this context, UAT is restricted to test the specified core user features. It does not include verifying the accuracy of the output (e.g. calculation, visualization), as these are determined through the functionality test.

UseR in Salzburg, XChen etc.

USER EXPERIENCE

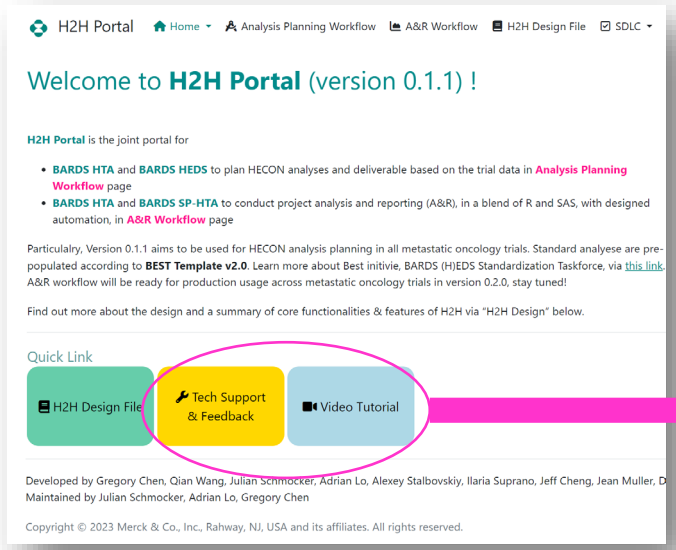
Software presents a solution to a business problem. **Key value** of a software is in user's experience on how to solve that problem with it (hence UX design at core)

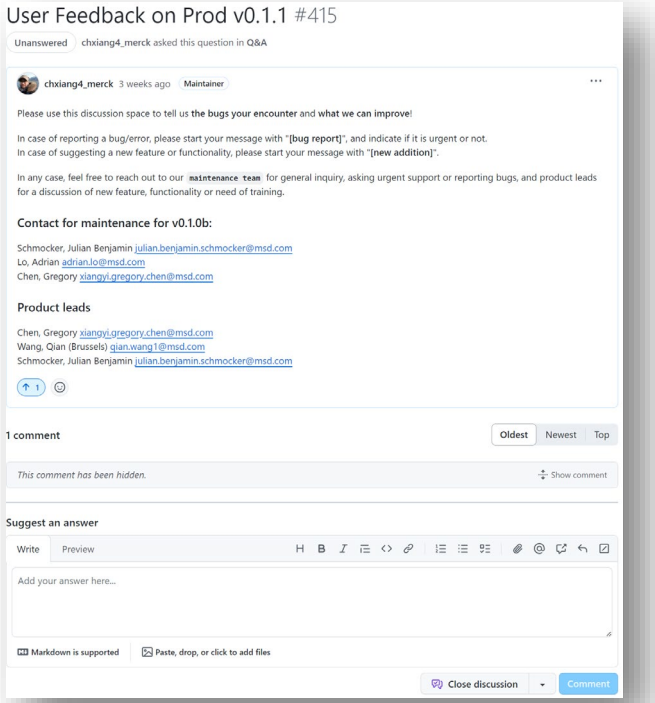
Incorporate or fine tune UI at UAT is too late, and not efficient

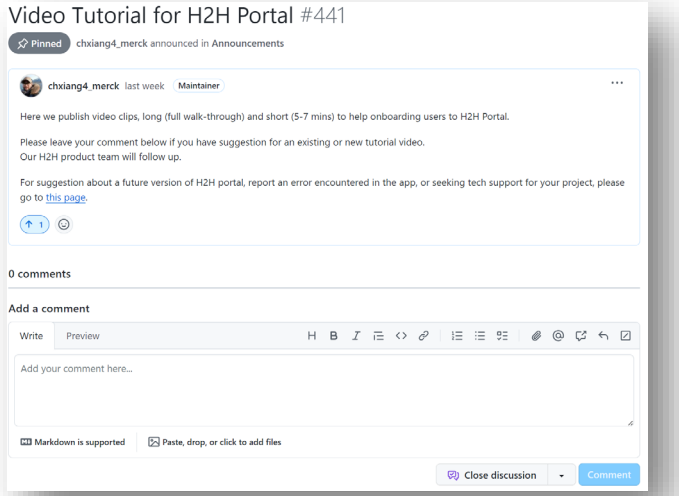
UAT is typically a set of sanity check, more evolved enough to properly collect all necessary user feedback in real world setting

SETUP TUTORIAL AND FEEDBACK CHANNEL

Dedicated channels in our **internal Github** for user feedback and training video library
(mostly short clips, each 5-7 mins, with informative title to facilitate user onboarding)







THANK YOU

Presenter: X. Gregory Chen

xiangyi.gregory.chen@msd.com

www.linkedin.com/in/xiang-yi-gregory-chen-5ba1a625/

Other Authors

Qian Wang

qian.wang1@msd.com

Julian Schmocker

julian.benjamin.schmocker@msd.com

Adrian Lo

adrian.lo@msd.com