

Analysis and Reporting Toolset (A&RT): Lessons on how to develop a system with an external partner

David Smith

AstraZeneca



Content

- Introduction
- Project Initiation
- Software development methodology
- Planning: Getting it right
- Collaboration: The importance of experts
- Sizing the application
- Implementation
- Lessons

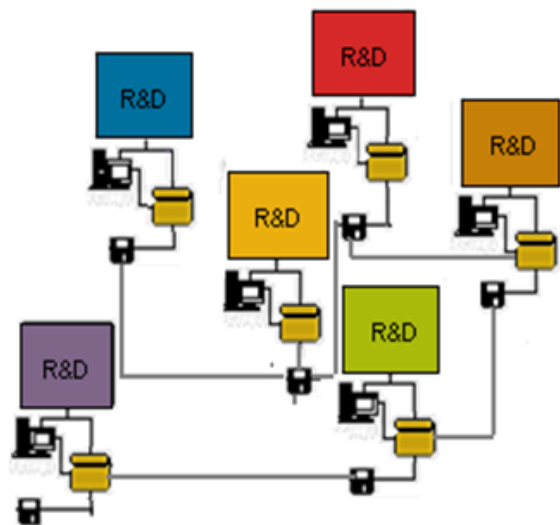


Introduction

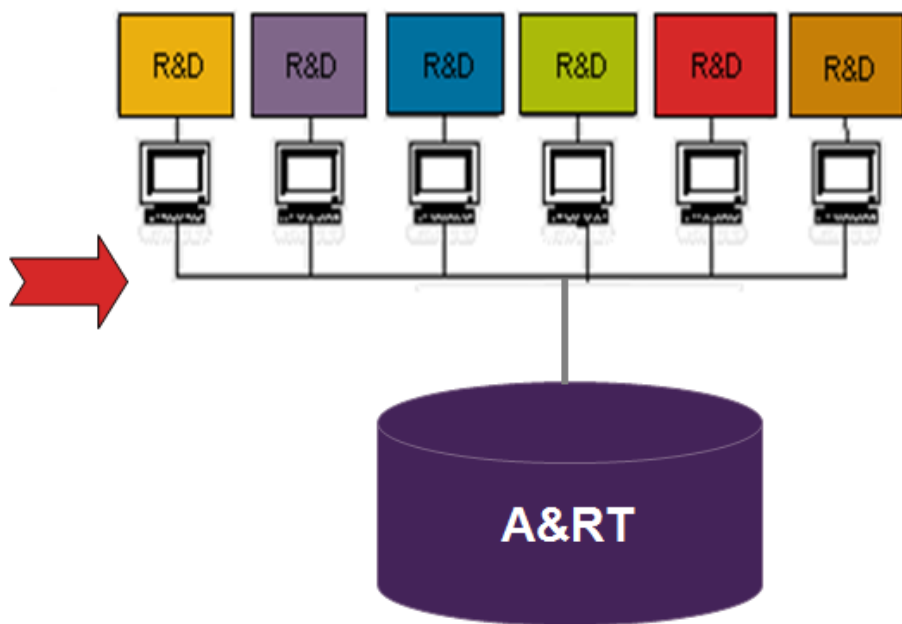
- For some years following Astra and Zeneca's merger clinical statistics and programming groups operated in a diverse way
- The A&RT project was setup to develop a standard analysis and reporting environment based on a set of integrated applications



Sharing A&R Assets The old AZ way



Sharing A&R Assets in A&RT

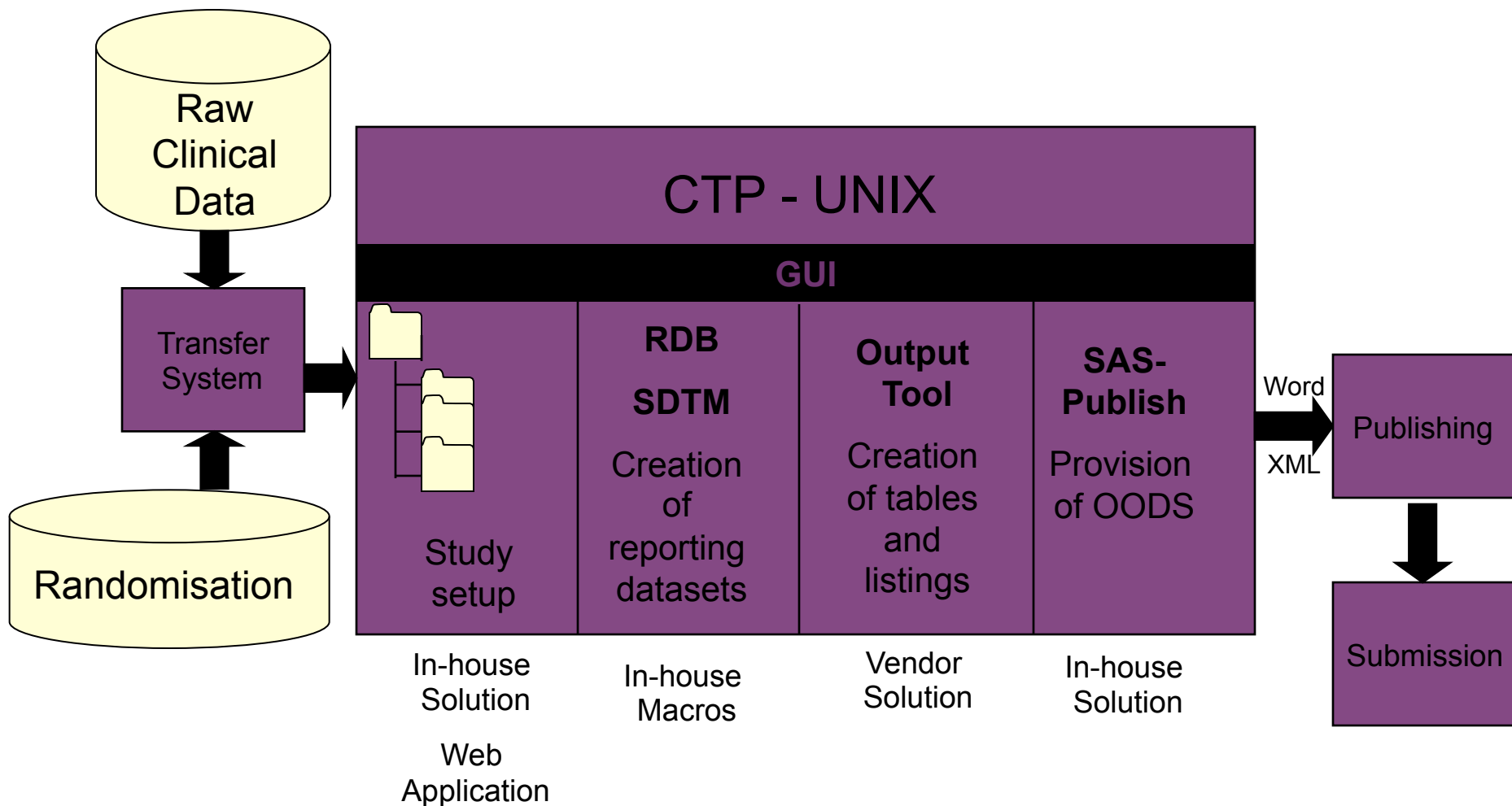


Project Initiation

- A complete User Requirement Specification (URS) was developed by the business
- A basic technical architecture was specified as part of the URS. This defined the requirements for
 - Powerful UNIX Platform
 - User interface
 - Data input, processing, reporting and publishing functionality



A&RT Technical Solution

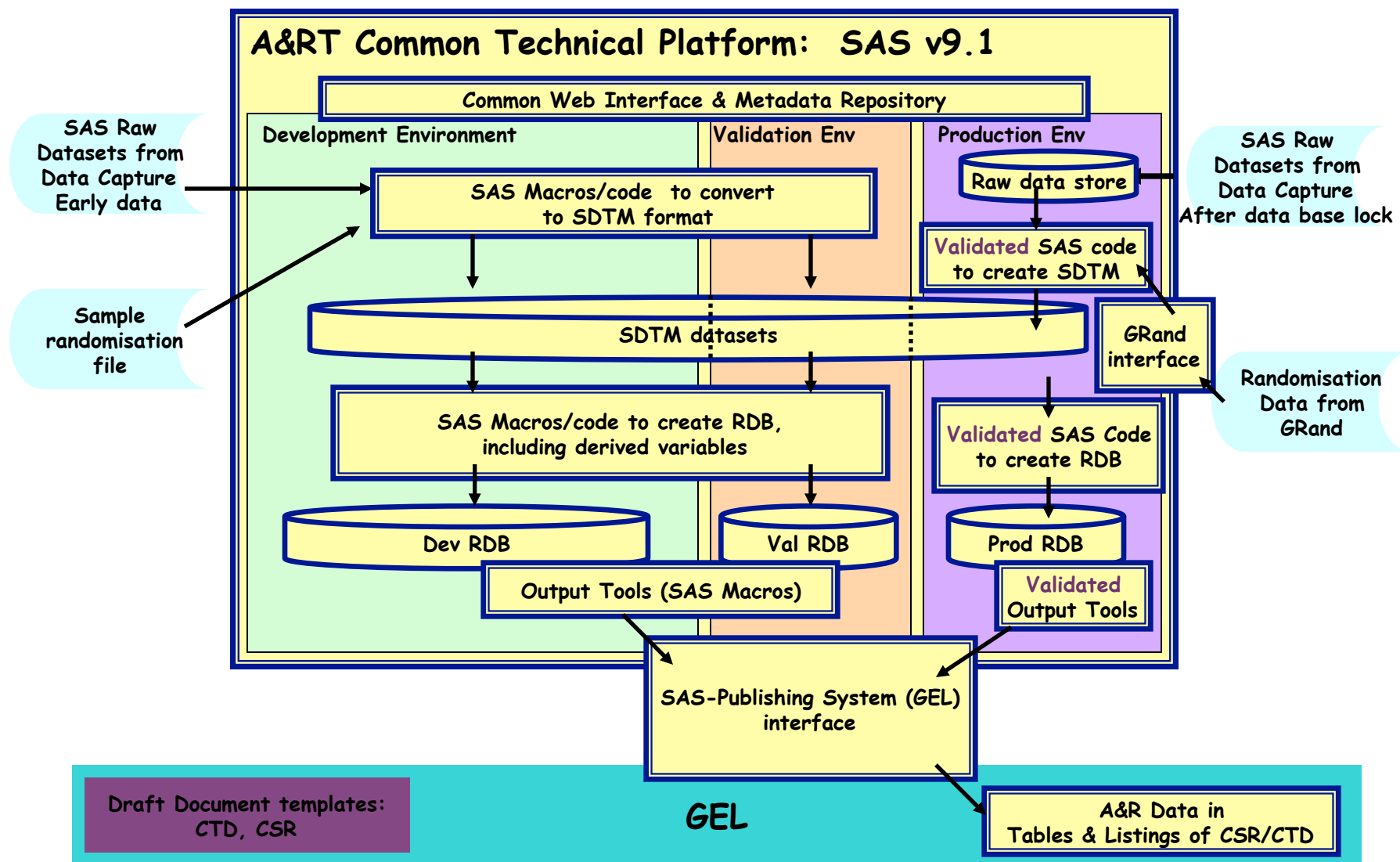


Project Initiation

- Functional business user requirements were also defined
 - Development, validation and production environments
 - Database folder structures for raw, SDTM and reporting databases (RDB)
 - Folder structure for SAS programs and outputs
 - Folder structure for publishing statistical tables and figures



Data flow through technical components...

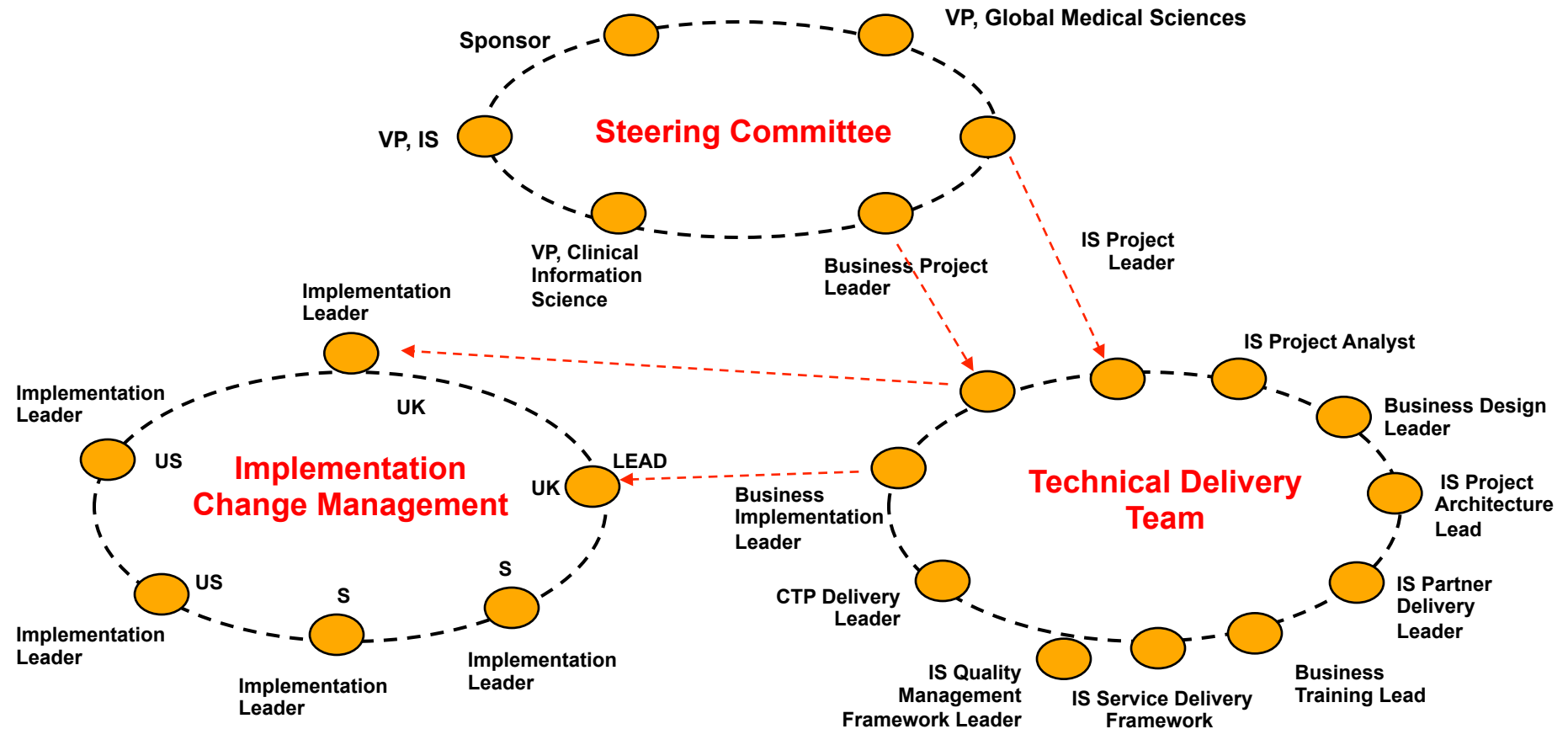


Project Team Organisation

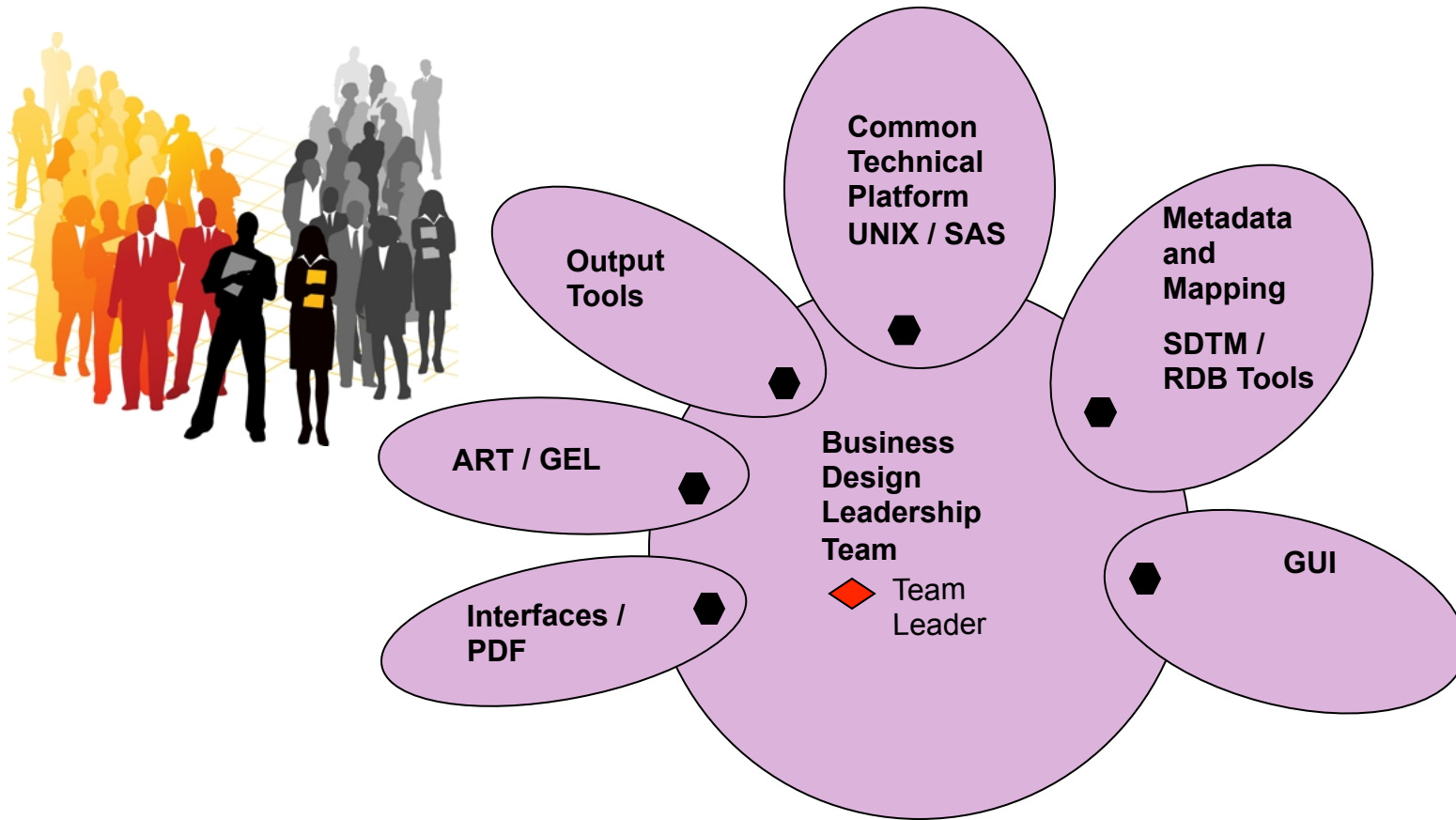
- A joint business and IS and business team was formed with a project leader from each function.
 - Controlled by a IS / business steering group
 - Leading a technical delivery team
- The formation of a Business Design sub team was a crucial part of the project
 - Provided the IS team with business application development expertise
- A separate implementation and change management team was formed
 - To manage drug project migration and process change



A&RT Project Governance Structure

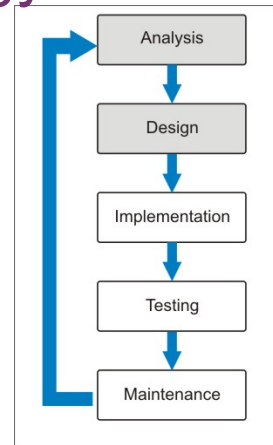


A&RT Business Design Organisation

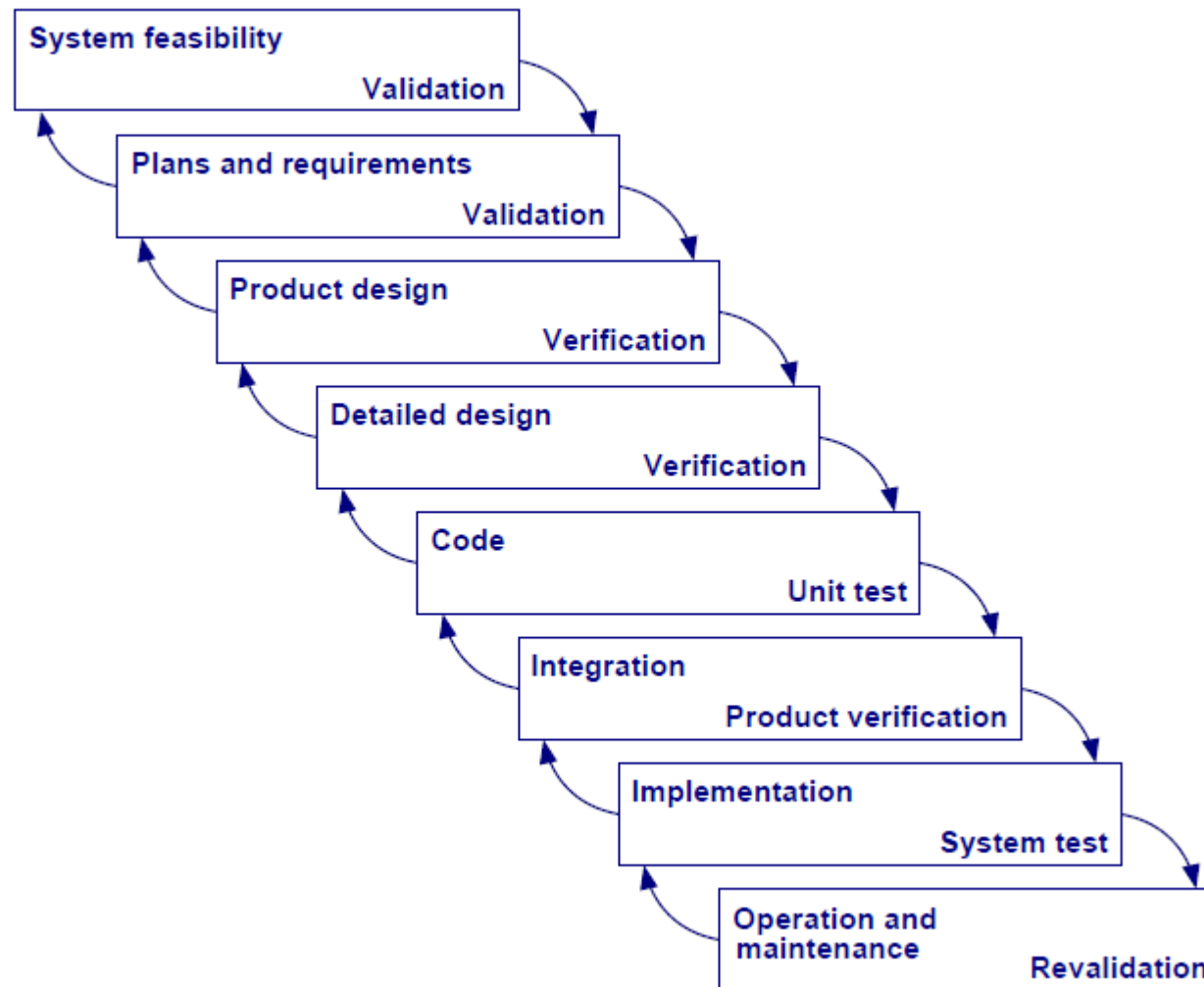


Software Development Methodology

- AstraZeneca IS followed a development methodology commonly known as the ‘waterfall’ method.
 - Also known as the Classic Life Cycle Model (or) Linear Sequential Model
- An alien methodology to the business application development programmers
 - More familiar with a flexible iterative prototyping methodology
- The business experts had to define precise and detailed functional requirements before any code was developed
- Formal sign-off required between each development step

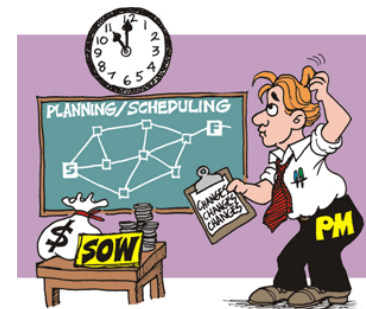


Waterfall Software Development Method



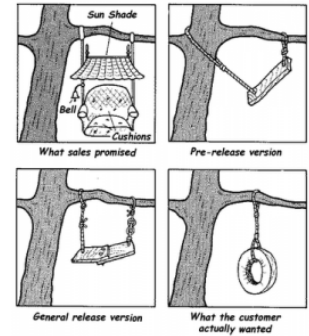
Planning: Getting it right

- Reality was not as easy as the waterfall theory suggested
- Thorough business analysis is required if the product design is to be sufficiently complete
- Missing detail at this design stage can have serious consequences later on
- The project team significantly underestimated the time required to develop the detailed design
- Despite a significant effort in time and manpower we still had to perform many iterations of the design specification after formal sign-off



Planning: Getting it right

- When we did receive our first test versions of the system significant problems arose
 - hardware infrastructure was not ready to run the applications built by the development partner
 - GUI screen and database functions did not work when loaded into the AstraZeneca environment
 - The partner delivered GUI and Oracle database that controlled the functionality of the system it did not work along side the AZ SAS servers
- The development partner had followed our design specification but AZ had not considered how this might be misinterpreted given the complex nature of clinical data

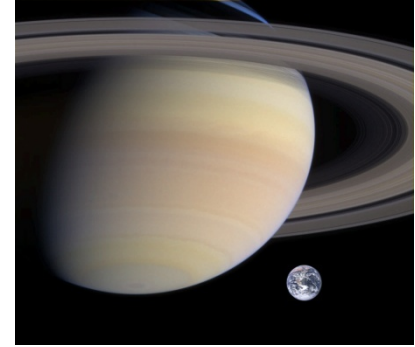


Collaboration: The importance of experts

- Mistakes in design and planning had to be corrected fast
- A rapid iterative development process was initiated
- A close team of business and development partner experts was formed
 - Business process experts
 - SAS application experts
- A shared development environment was created
- Domain knowledge of the partner was significantly improved
- This expert to expert collaboration continued through development, testing and implementation



Sizing the Application



- The first release of A&RT appeared to function well until the user base started to expand
- Issues were soon detected that had gone unnoticed during the formal system and user acceptance testing
- What works for 5 studies and 10 users did not work for 100 studies and 200 users
- In the haste to deliver, adequate performance testing had not been performed
- we realised too late how critical formal load testing was to the final success of the tool

Implementation – Success?

- A&RT was delivered into full production use in 2009
- Functionally the system did what it was designed to do so this part was a success
- Still an ongoing challenge to fully deliver the system performance users expect. This is the subject of ongoing system enhancements based on load testing performed late in the project.



Lessons (1)

- Before embarking on a complex system build consider the possibility that a vendor ‘off the shelf’ supplied solution might be more cost effective
 - Compare expenses that include all in-house resource use over an extended time
- Resist the desire by IS to follow a waterfall approach
- Business experts must be part of a prototype review team
- Follow the agile software development methodology based on iterative and incremental development
- Business and IS development experts must work together as a team



Lessons (2)

- IS developers must have access to experts e.g. SAS application experts, Oracle experts and system performance experts.
- Plan for a number of prototype iterations to be performed
 - Recognise resource will be required for an extended period
- Project planners necessarily have an eye on budgets and timelines but recognise that more time may be needed in reality to ensure robust testing.
- Include non-functional requirements describing expected user and data load on the system along with the expected performance
- Always include load testing



Questions

