

How specifications get lost in translation

Ralf Minkenberg

Contents



- Introduction
- Problem examples
 - How Programmers are lost at the end of a long chain with specifications and discussion
- Problem solutions
 - How Programmers can give valuable input into the development of specifications
- Conclusions

Daily Life of a Programmer

What should be done?
How should it be done?

Definitions
Specifications



Details:

- Tasks
- Data preparation
- (Statistical) analyses
- Results presentation
- . . .



Are there any problems?

Request

Misunderstanding

Incomplete

Data

Wrong sources

Missing derivations

Analysis

Incorrect procedures

Biased . . .



Review – Comments – Corrections –
Expansions – . . .

Different involved functions

At the end – the Programmer

Idea of request

Involving other functions

Draft specifications

Final Specifications

Discussion



Discussion

Discussion



Discussion (?)

----> Chinese whisper chain with Programmer at the end ----> What is really wanted?



Example 1

Ad-hoc request about Adverse Events

- caused by journal article
- initiated by Safety Dep
- time critical
- forwarded to Statistician
- creates detailed specs

Everything clear?

- data sources / access
- coding version (MEDDRA)
- special search categories



Example 2

Trial results are discussed on management level

- need for additional analyses
- many functions are involved
- (therapeutic area, Medical Affairs, Stats, Drug Safety, ...)
- analysis suggestions in final minutes

Everything clear?

- technical specs missing
- ambiguous specs in minutes
- forth and back between Prog, Stat, and requestors



Summary

Origin of requests

- medical driven questions
- not familiar with our problems

Different ideas sum up to one request

- ambiguous, unclear definitions
- different background

Programmer not involved in decision process

- contents of request completely lost
- unknown, many interfaces

What could be changed?

Essential: background knowledge



Involve Programming as early as possible

Equal partners with Stats, Medics, DM, ...

Interdisciplinary interaction



Prerequisites for Programmers

Good programming skills

Good knowledge of basic statistical methods

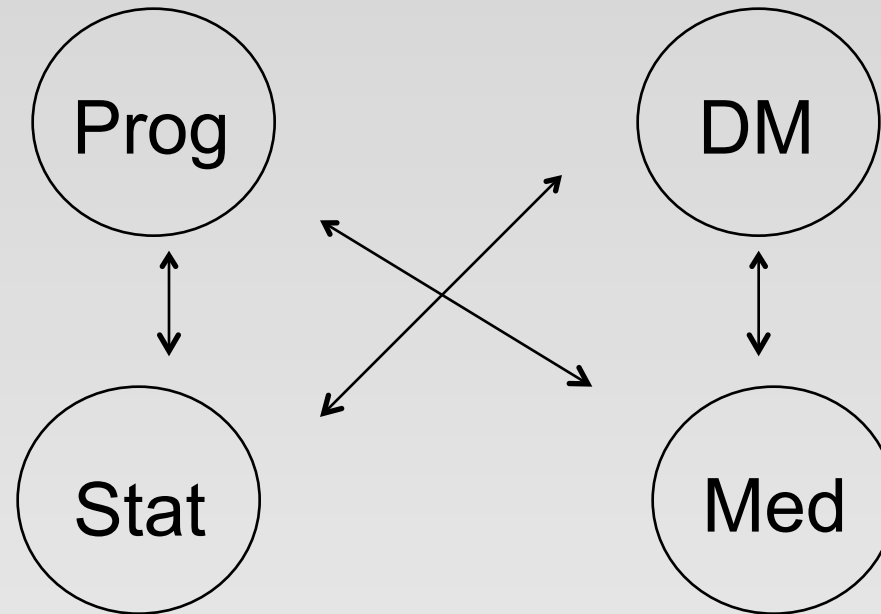
Informed about basic therapeutic concepts

Multidisciplinary background

Life-time training in SAS, Statistics, and Medicine

Conclusions

Interaction between all involved functions is the key



Participation of all functions from start to end

Thank you

