Development of SAS Standard Macro Systems

Katja Glaß, 10. October 2011, PhUSE
Contents

- Introduction
- Macro System Paradigms
- Specifications
- Implementation and Rollout
- Additional Tips
- Summary
- References
Introduction

- Productivity, Quality
- Standardized programs / results
- Reduced validation effort
- "Source generator"
- Commercial software
Macro System Paradigms - Purpose

- "Toolbox" - bunch of tools
  - Daily needed supporting tools
- "closed" macro systems
  - Few parameters with low complexity and flexibility
- "open" macro systems
  - Many options with high complexity and flexibility
Macro System Paradigms - Types

- SAS Macros called in SAS
- SAS Macros called by a GUI
- SAS Macros called via Web

- User groups
- Efficiency
- System confidence
Macro System Paradigms - Developer

- In house development
  - Company knowledge, workflow, processes, cases
  - Motivation, Acceptance

- External development
  - Expertise, efficiency for documentation, implementation and validation
  - Less user-friendly, enhancement barriers
Macro System Paradigms - Models

- Design Models
  - Waterfall Model
    (Requirements, Design, Implementation, Verification, Maintenance)
  - V Model
    (Concept, Architecture, Detailed Design, Test & Verifications)
  - Rapid Prototyping
  - Agile Programming
    e.g. Extreme Programming (XP)
    (Frequent Loops)
Macro System Paradigms

- Extreme Programming (XP)
  - Frequent releases
  - Short development cycles
  - Changing customer requirements
  - High user acceptance
  - Quick wins

Figure 1: Extreme Programming
Specifications

How the customer explained it

How the analyst designed it

What the customer really needed

Figure 2: Specifications
Specifications

- Who is creating specifications (specifications via group)
  - Technical / content knowledge
  - Group size
    - Small $\rightarrow$ efficient
    - Large $\rightarrow$ comprehensive
  - Types
    - Enforced $\Leftrightarrow$ pleased
    - Skills
    - Functions
    - Empowered
Specifications

- How is the specification
  - Technical, content wise
  - Detailed, strict, interpretation open
  - Exception handling

Less is more, but not enough!
Implementation

- Phases and motivation during implementation
Rollout – Background of Users

Daily work

Training sessions Learning Curve
Rollout – Background of Users

- Open for changes
- Accept changes
- Conservative
Rollout – Motivation of Users

- User-friendly
- Support, training, documentation, intuitive parameters
- From-Ourselves-For-Ourselves
  - Implementation / Specification
- Embedding in enhancement cycle
  - Change request lists (easy to address)
- Confidence into the system
  - No black box
- “Fun” part
- Management support
  - Enable / force learning time
Additional Tips

• Collect any idea
  - Could result in significant improvements

• New ideas by students
  - Work experiences, bachelor and master theses
  - Recruitment support

• Focus on users
Summary

- General principles (SAS macros, GUI, inhouse, …)
- User as developer
- Teams as small as possible
- Rapid Prototyping for fast benefits
- One-Person-Projects
- Continues improvements through user ideas
- Focus on Users
References

• Figure 1: http://en.wikipedia.org/wiki/File:XP-feedback.gif by DonWells 1 May 2001
• Figure 2: http://www.projectcartoon.com/cartoon/2

Share your thoughts on PhUSE Blog!!!
Make your users happy!!!