



# SCRUM

in a waterfall

ScrumDay '09, Thilo Fromm

## The Waterfall Process

General concept and DResearch's Implementation

## Agile Development

A quick review

## Best of both worlds

Making it all work together

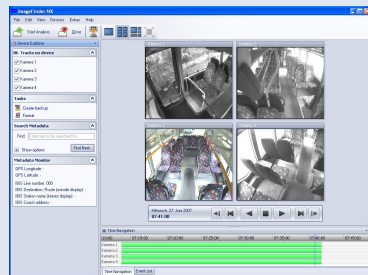
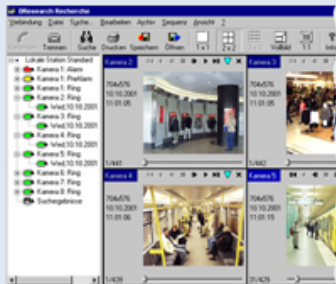
# Our Business

Mobile DVRs for security applications

Public Transport Sector

Whole Products, Product Lines

ISO 9001-2000



# The Waterfall Process

Traditional approach for systems design

Big Design Up Front (BDUF)

Widespread use

Certification friendly

# The Waterfall Process

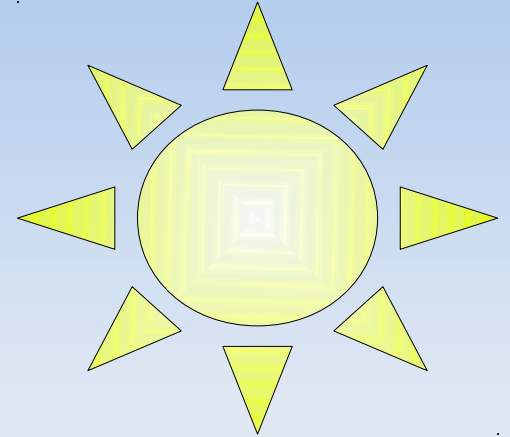
Requirements

System Design

Implementation

Verification

Maintenance



# Waterfall Process - DResearch

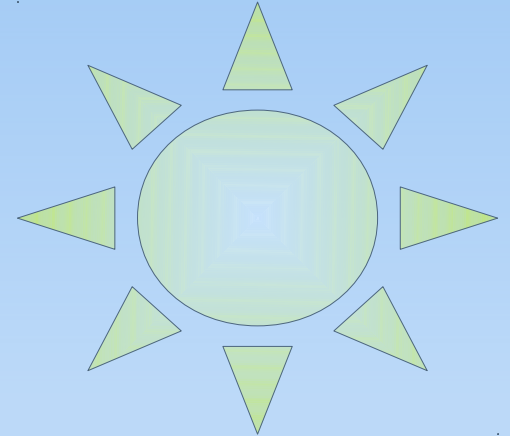
Requirements  
*Lastenheft*  
*Pflichtenheft*

System Design

Implementation

Verification

Maintenance



# Waterfall Process - DResearch

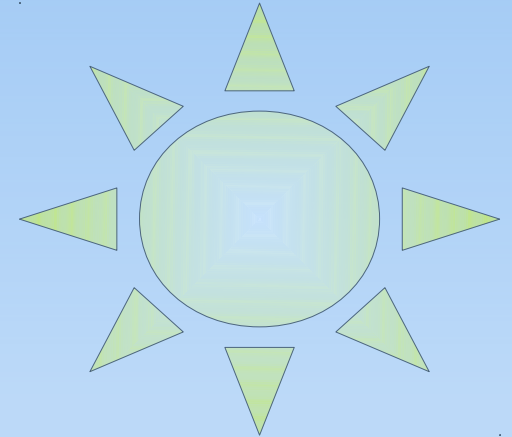
Requirements  
*Lastenheft*  
*Pflichtenheft*

System Design  
*Design Specs*

Implementation

Verification

Maintenance



# Waterfall Process - DResearch

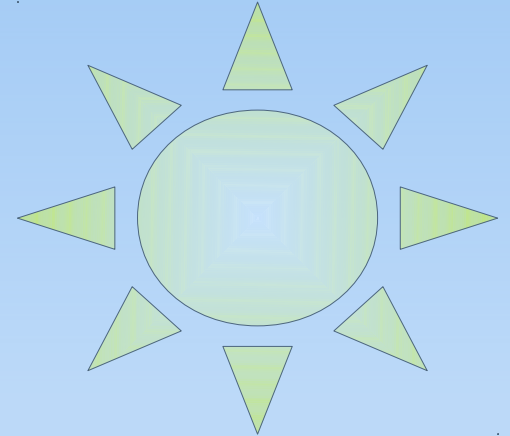
Requirements  
*Lastenheft*  
*Pflichtenheft*

System Design  
*Design Specs*

Implementation  
*Functional Milestones*

Verification

Maintenance



# Waterfall Process - DResearch

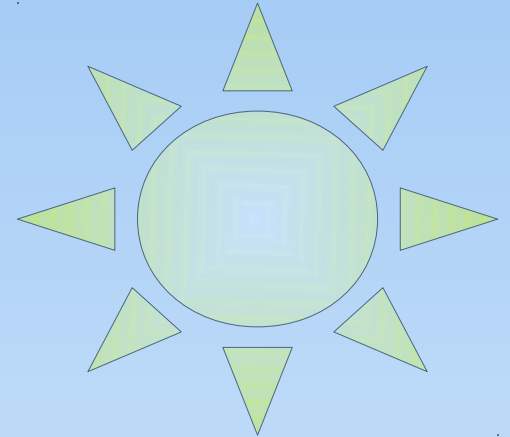
Requirements  
*Lastenheft*  
*Pflichtenheft*

System Design  
*Design Specs*

Implementation  
*Functional Milestones*

Verification  
*Test Specs + Milestones, Bugfixing, Release Tests*

Maintenance



# Waterfall Process - DResearch

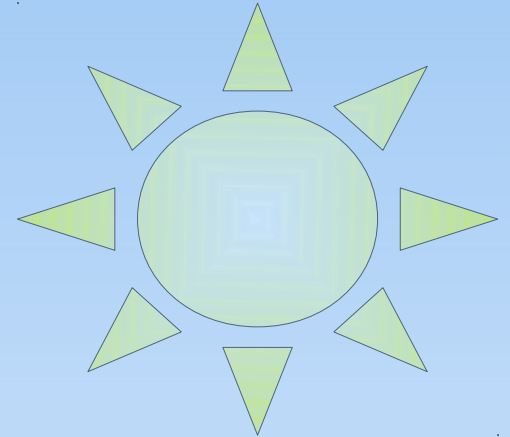
Requirements  
*Lastenheft*  
*Pflichtenheft*

System Design  
*Design Specs*

Implementation  
*Functional Milestones*

Verification  
*Test Specs + Milestones, Bugfixing, Release Tests*

Maintenance  
*Feature Request, Change Request tracking*



# Waterfall PROs

Structured, step-by-step approach

“You get what you ordered”

Documented prior to development

Long term plannable

Cost, time, Features are fixed

# Waterfall CONs

Requires throughout Requirement / Design

- Customer communication issues

- "Ivory Tower" designs

Incredibly long round-trip times

- Development set for months / years

- Slow reaction to requirement / market changes

No built-in reflection, no rethinking

# Waterfall @DResearch

Involved Parties and Roles

Communication flow

# Requirement Engineering



Project Engineers



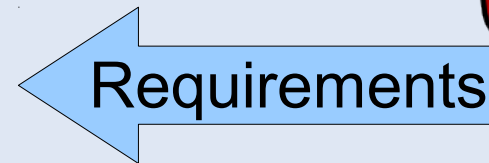
Test Lab



Project Lead



Product Manager

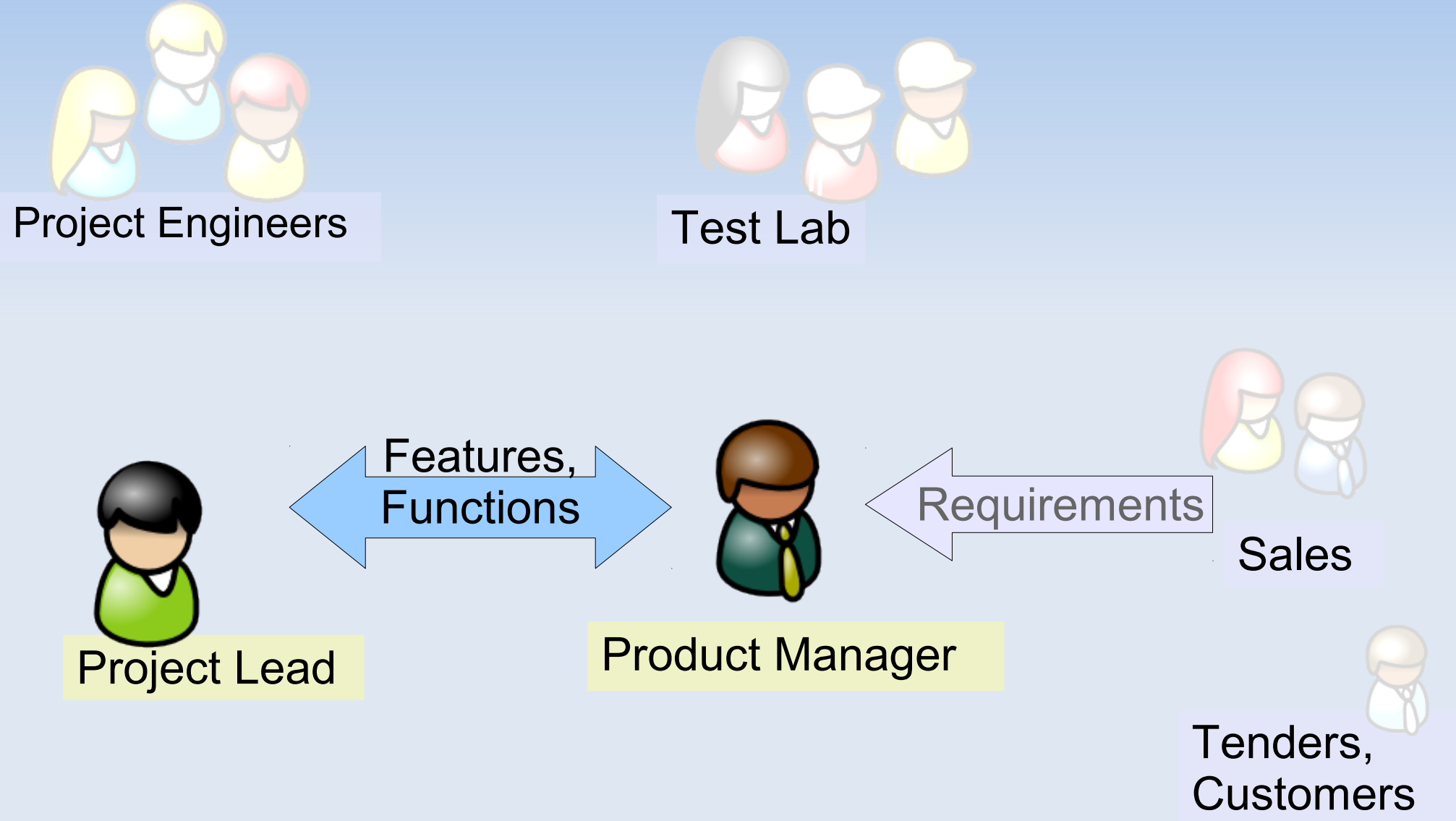


Sales

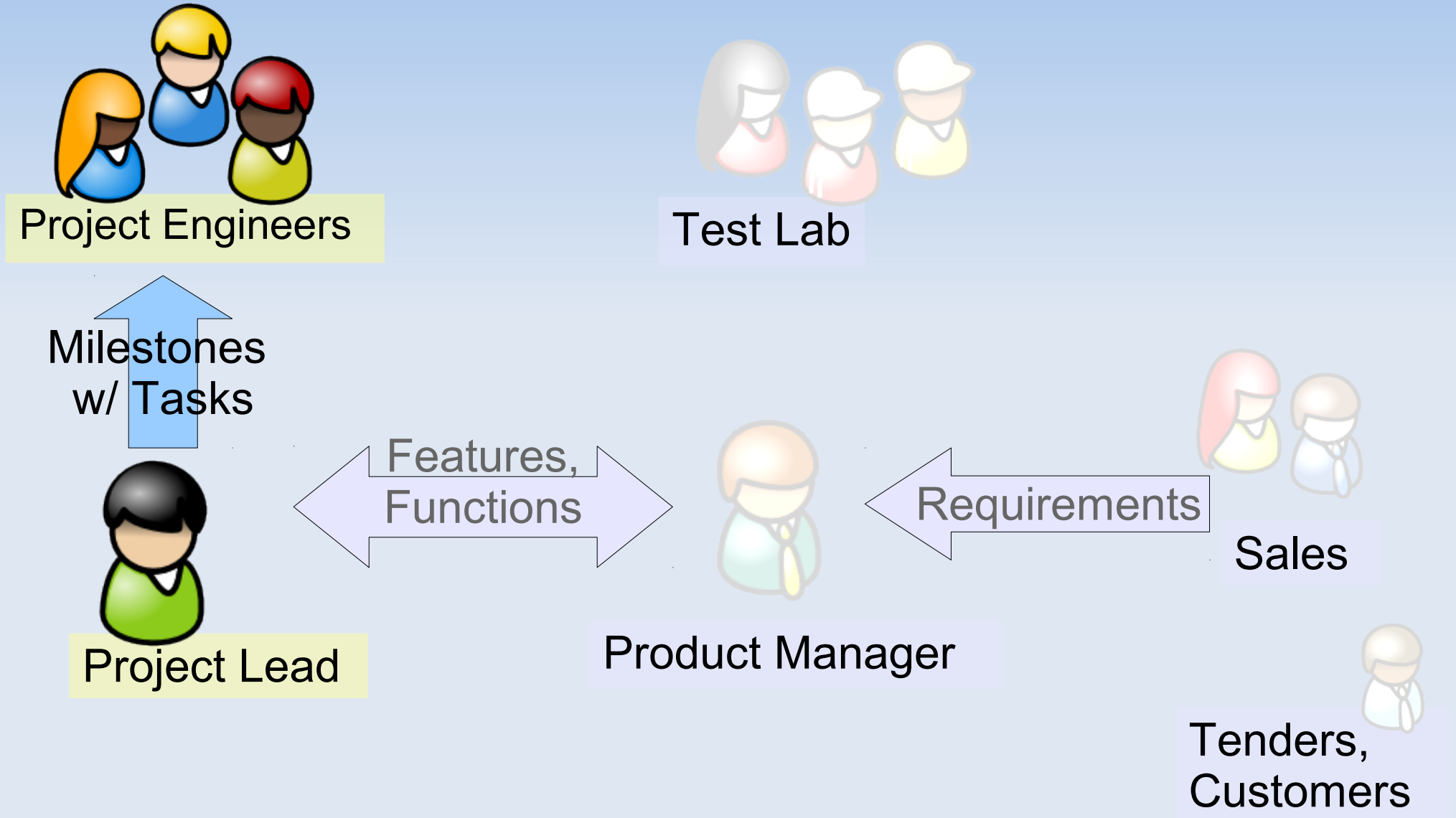


Tenders,  
Customers

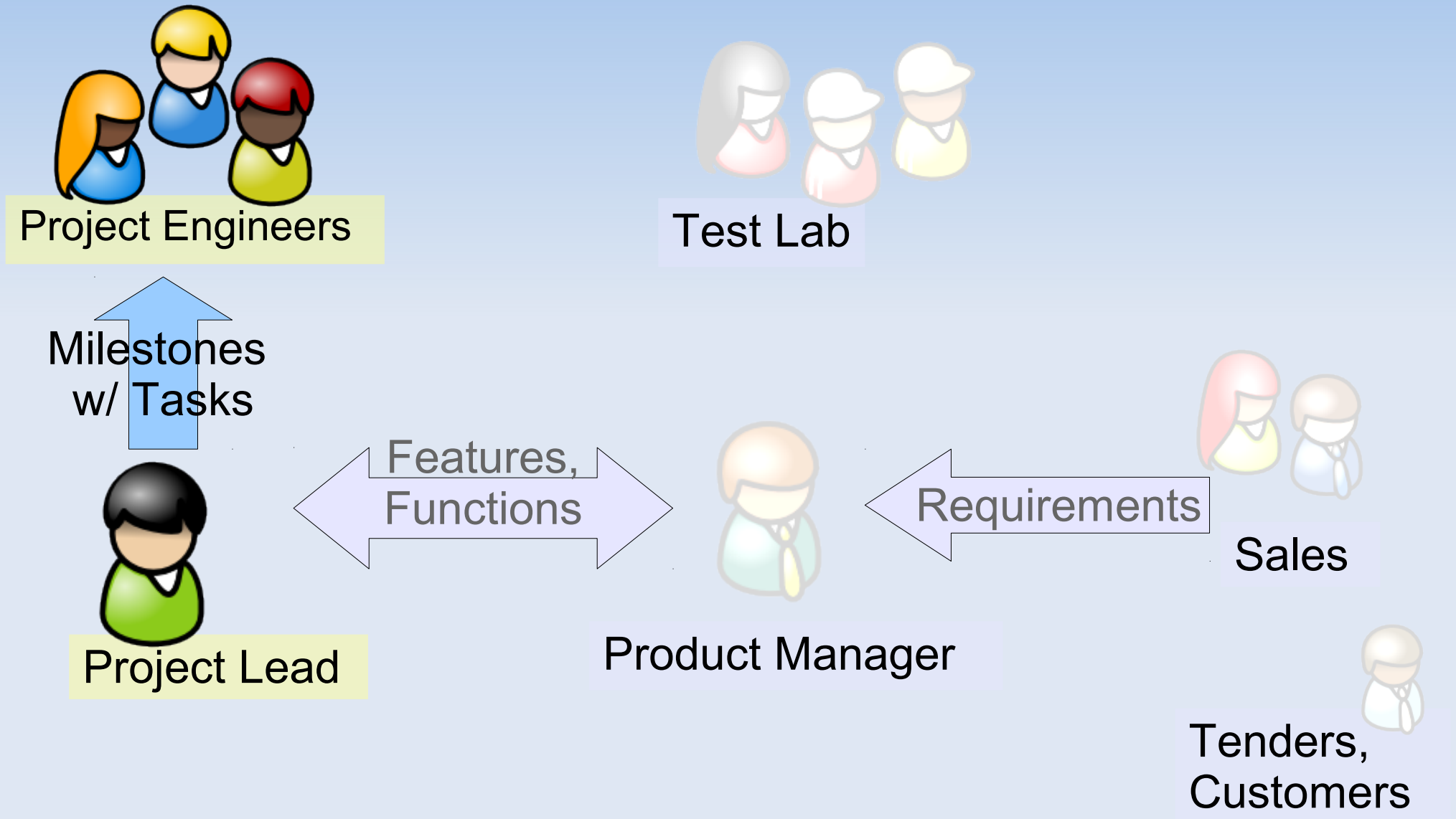
# Lasten- / Pflichtenheft dialogue



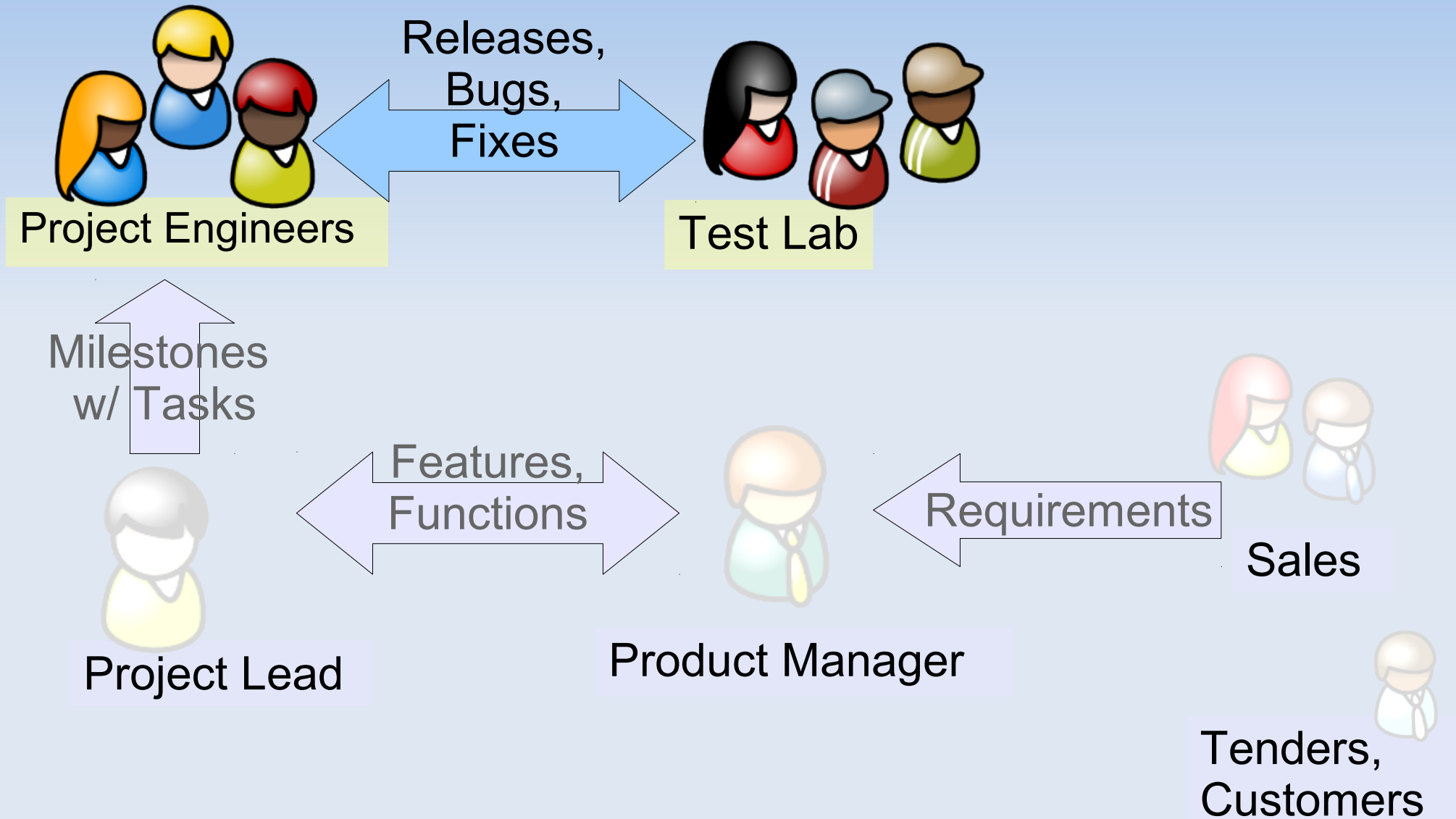
# Design



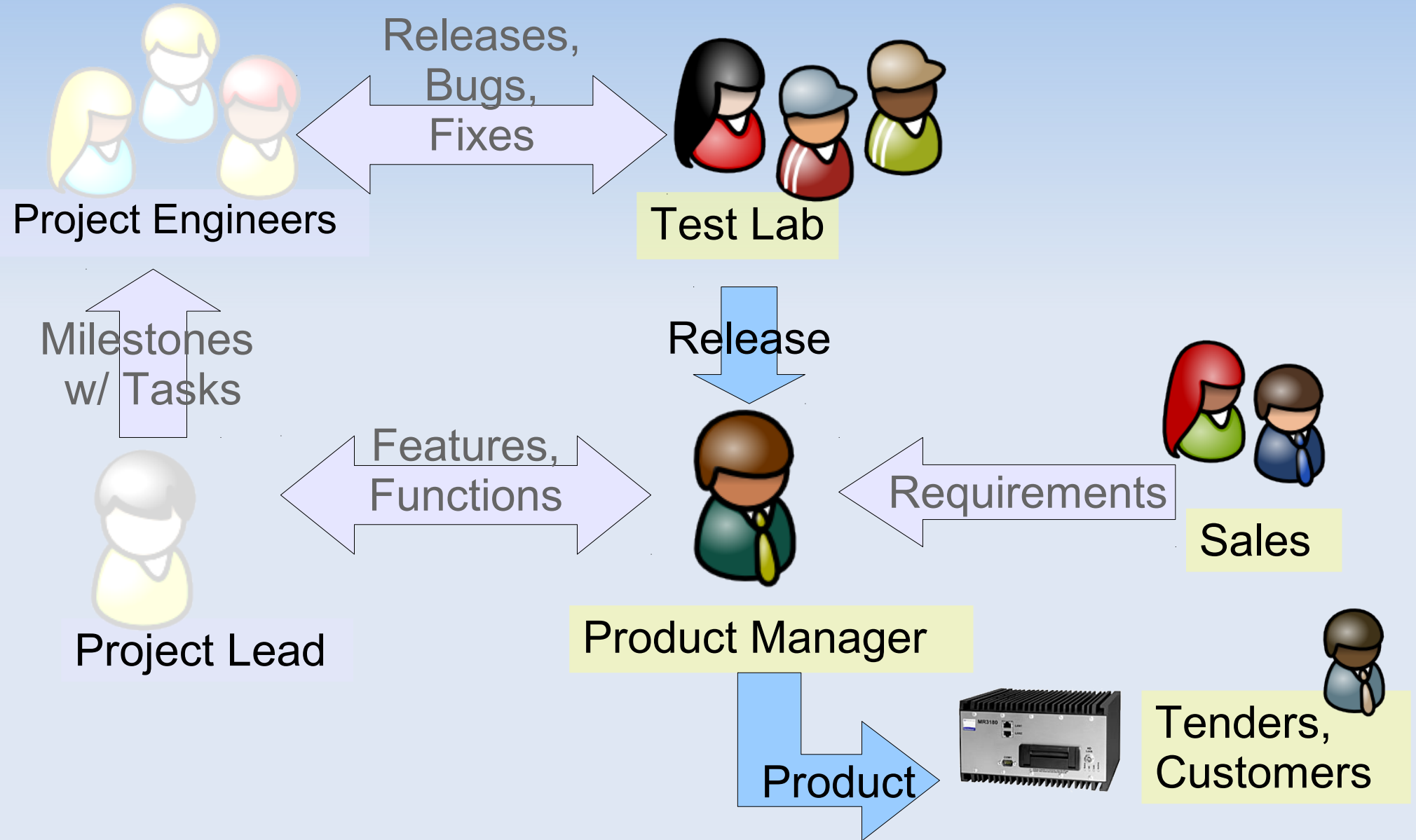
# Implementation



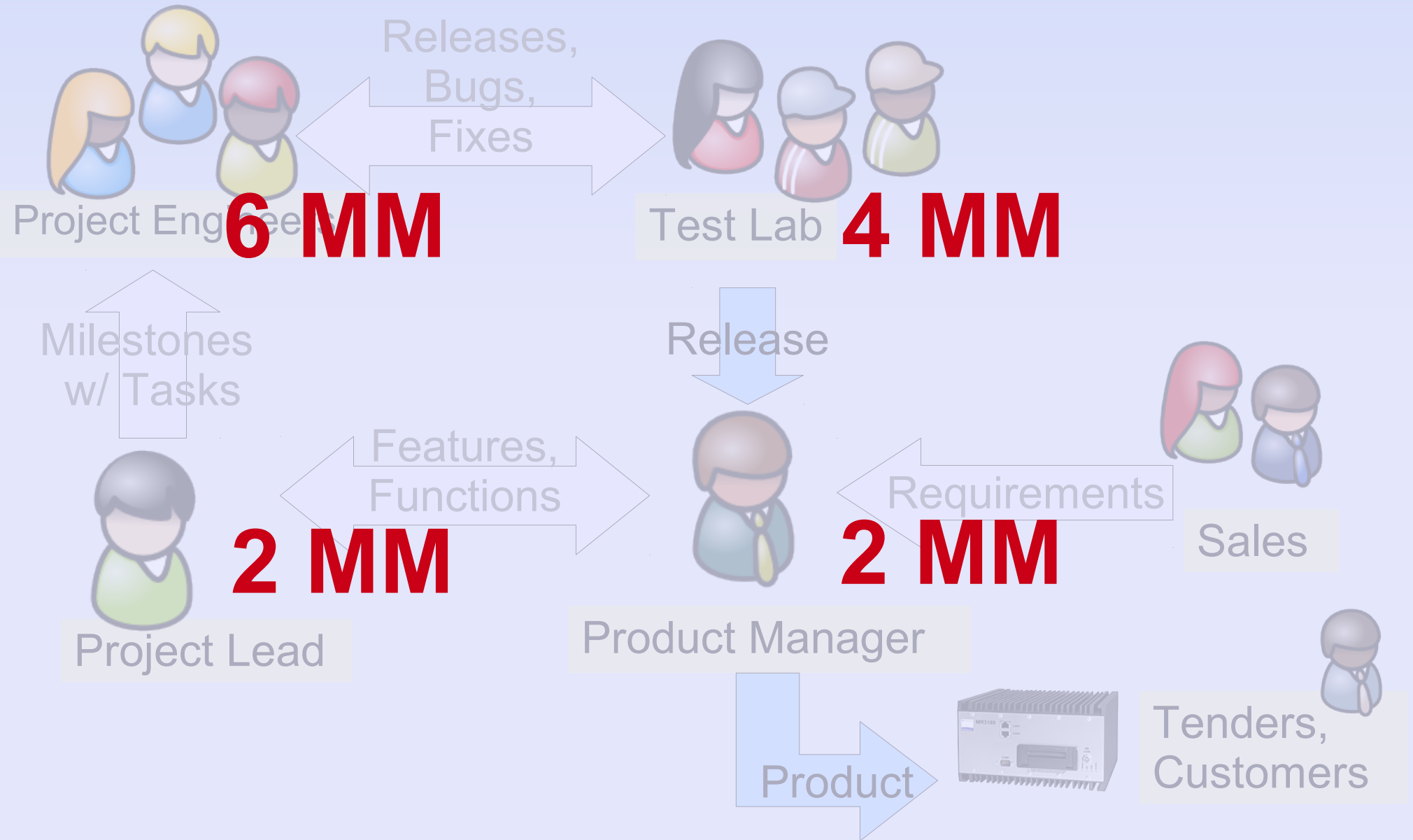
# Verification



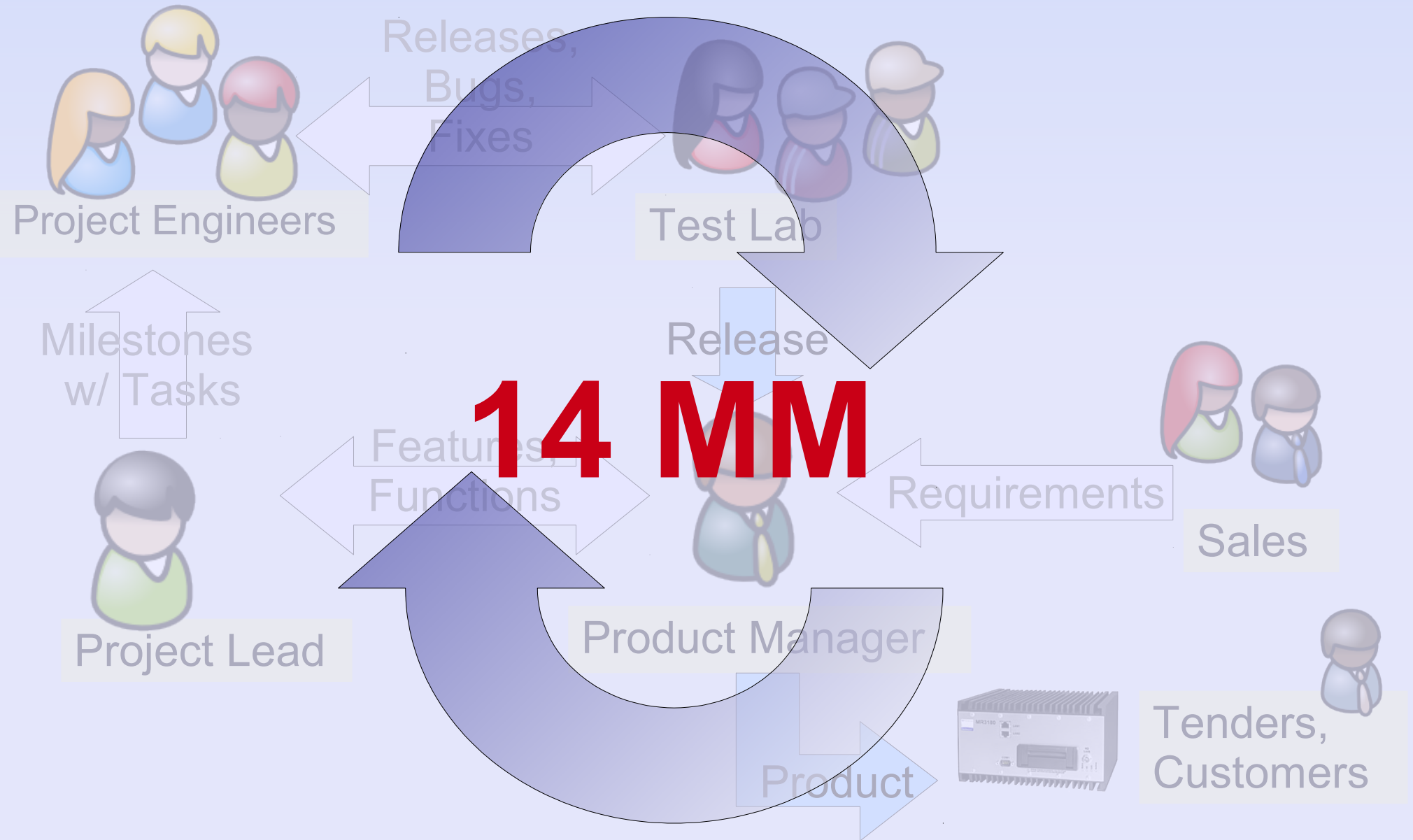
# Release



# Release



# Release



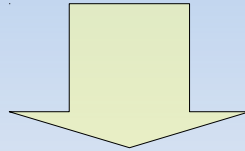
# Implementation / Verification

In-depth implementation steps

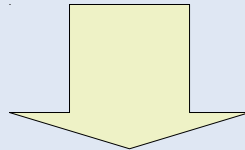
Repeated verification phases

# Implementation - DResearch

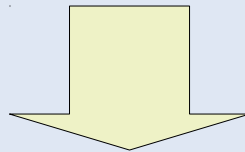
Milestone #1: "Recording"



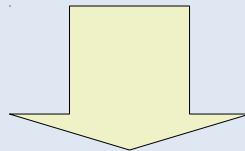
Milestone #2: "Configuration"



Milestone #3: "Live View"



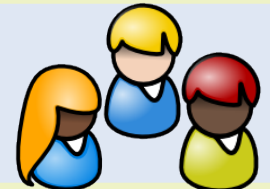
Milestone #4: "Image Search"



...

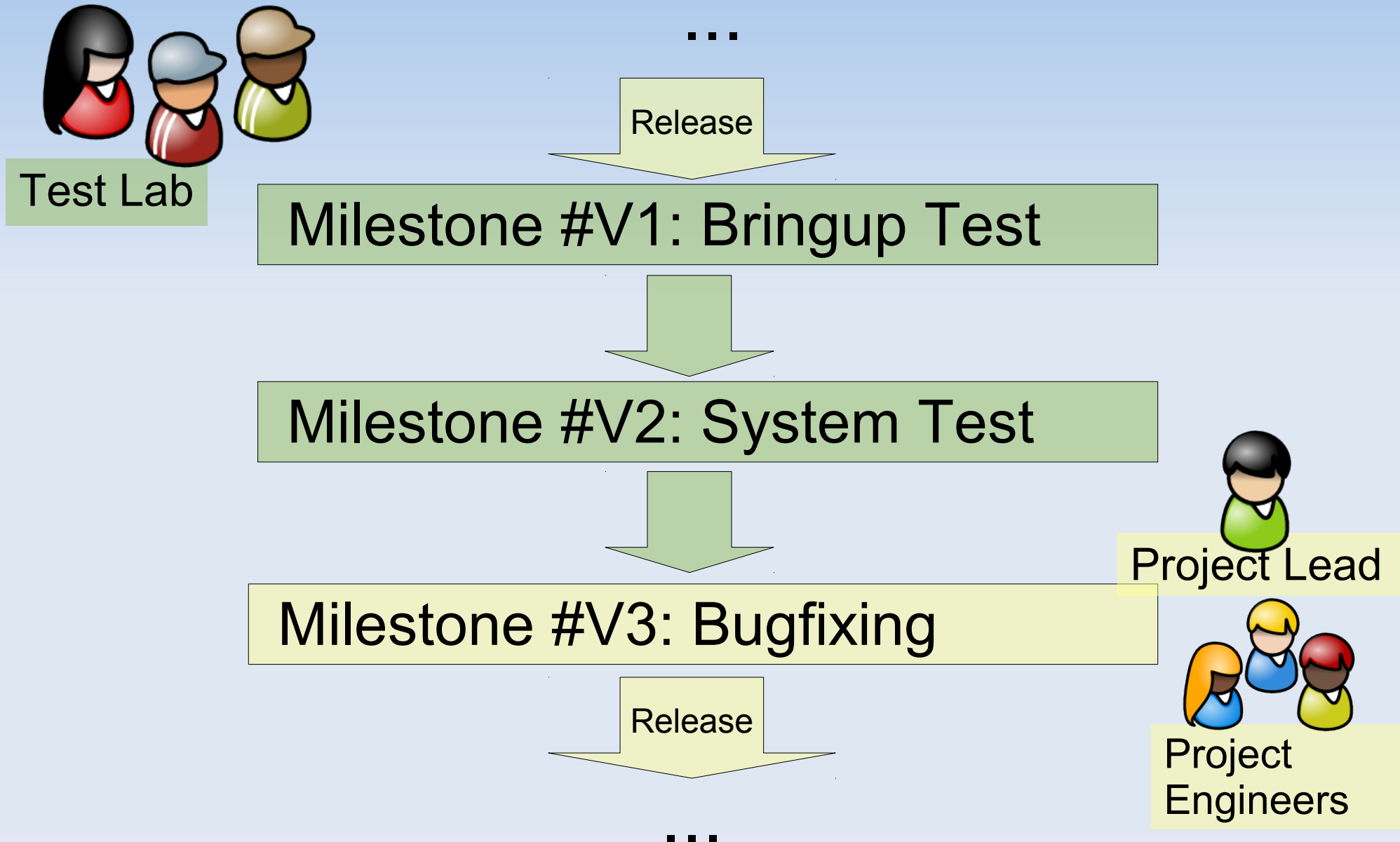


Project Lead

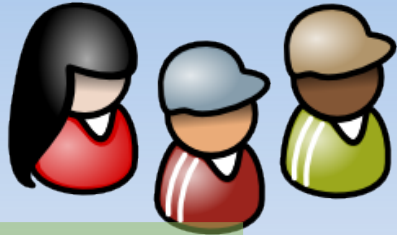


Project Engineers

# Verification - DResearch



# Verification (contd.) - DResearch



Test Lab

...

Release

Milestone #V4: System Test II

Milestone #V5: Bugfixing II

Release

Milestone #V6: Release Test



Product Manager



Project Lead



Project Engineers



# Process deficiencies

## Long TTM (~ 6 mths – 1.5 yrs.)

- Feature ROI hard to determine

- Missed project opportunities

- Sales orders all-inclusive products

## Inefficient verification

- Bug detection → fix → validation RTT ~2-4 weeks

- Loss from specific bug hard to determine

- 2-4 mths. verification for each release

# Striving for Improvement

## Known deficiencies

Are there known solutions?

## Extensions to Waterfall

What do we need to change, why and how?

## Alternative processes

SCRUM, Agile, Lean, – what about ISO-9001?

# SCRUM / Agile Methods

Development phase centered approach

Work in small, incremental steps

Verify everything you do

Design as you go

Release early, release often <sup>TM</sup>

# SCRUM / Agile Methods



Product Owner

## User Stories

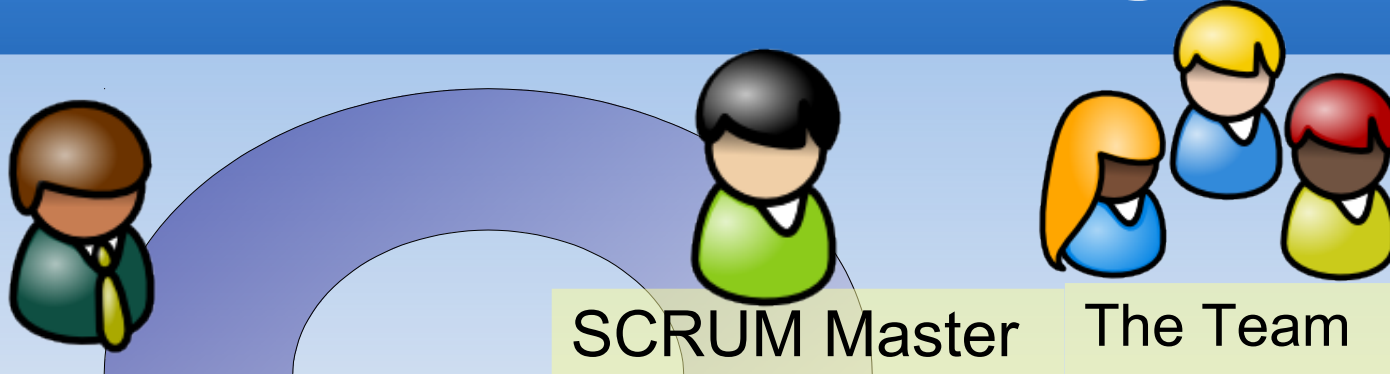
Feature #1 ...

Feature #2 ...

Feature #3 ...

...

# SCRUM / Agile Methods



Product Owner

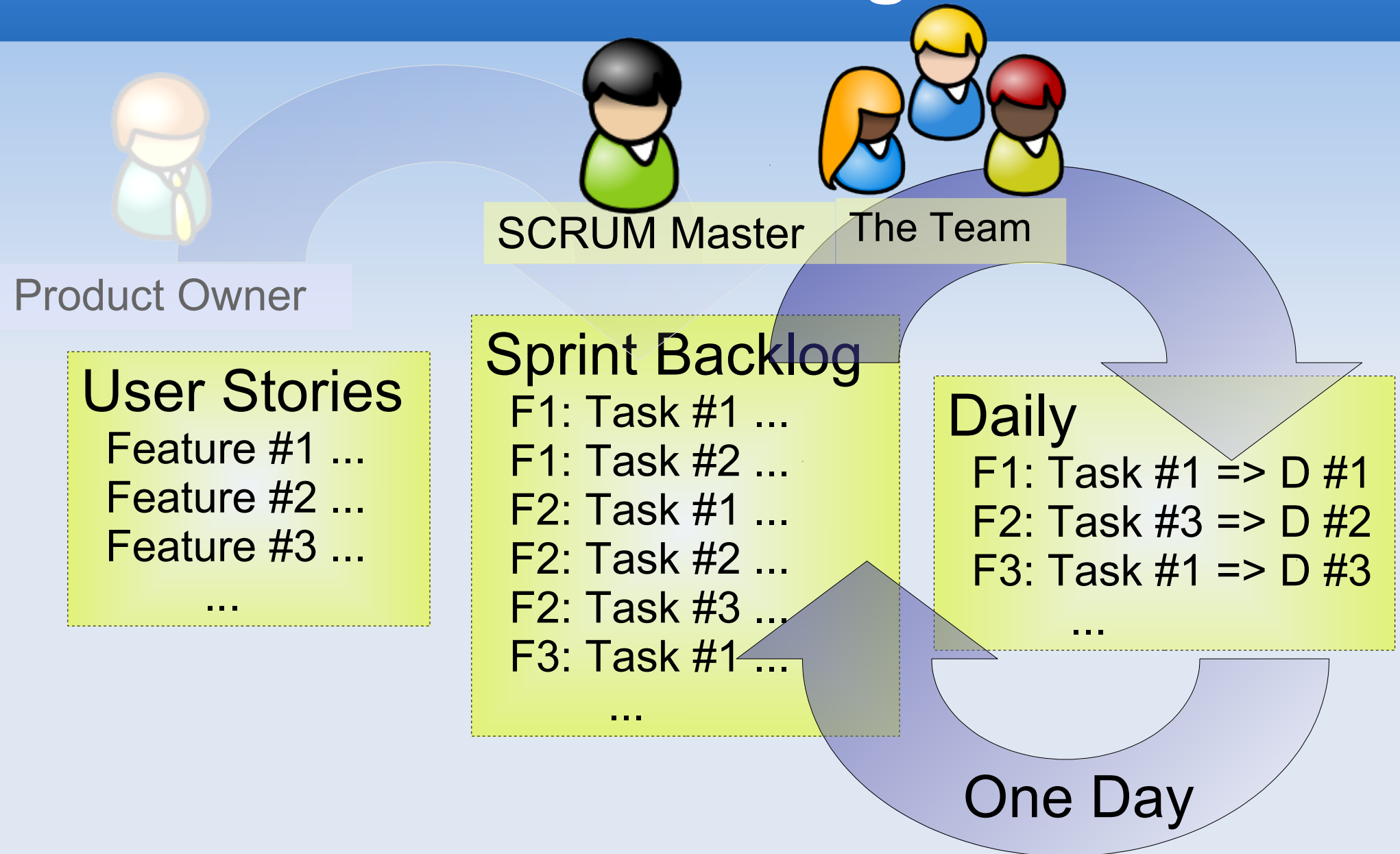
## User Stories

Feature #1 ...  
Feature #2 ...  
Feature #3 ...  
...

## Sprint Backlog

F1: Task #1 ...  
F1: Task #2 ...  
F2: Task #1 ...  
F2: Task #2 ...  
F2: Task #3 ...  
F3: Task #1 ...  
...

# SCRUM / Agile Methods



# SCRUM / Agile Methods



Product Owner



SCRUM Master



The Team

## User Stories

Feature #1 ...  
Feature #2 ...  
Feature #3 ...  
...

## Sprint Backlog

F1: Task #1 ...  
F1: Task #2 ...  
F2: Task #1 ...  
F2: Task #2 ...  
F2: Task #3 ...  
F3: Task #1 ...  
...

## Daily

F1: Task #1 => D #1  
F2: Task #3 => D #2  
F3: Task #1 => D #3  
...

One Day

2-4 Weeks

# Agile PROs

Highly flexible approach

Optimize for TTM, cost, or personnel available  
**as you go**

“You get what you need most”

Built-in rethinking, reflection

# Agile CONs

## Global concept not part of the process

Global design relies on Master and Owner

Unbalanced end products

”Mad cow featuritia”

## Depends on Customer behaviour

C. must be willing to participate in design

C. must be available

# Joining Methodologies

## Waterfall advantages

- Structured, plannable approach

- Big picture is communicated and documented

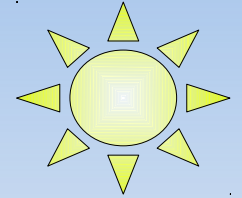
## Agile advantages

- Flexible in features, cost, TTM, personnel

- Focus on market value

- Early releases

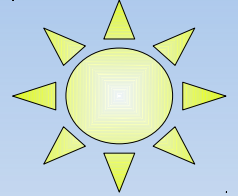
# Joining Methodologies



Requirements

Design

# Joining Methodologies



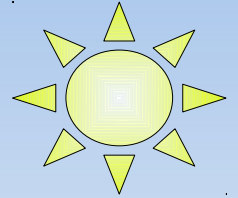
Lightweight Requirements

*Lastenheft*

*Pflichtenheft*

Design

# Joining Methodologies



Lightweight Requirements

***Lastenheft***

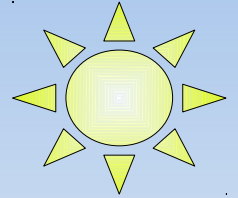
***Pflichtenheft***

Lightweight Design

***Design Specs***

*(where necessary)*

# Joining Methodologies



Lightweight Requirements

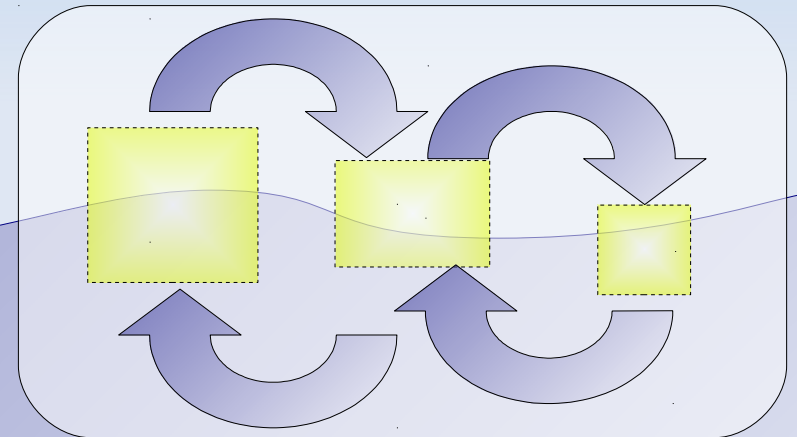
*Lastenheft*

*Pflichtenheft*

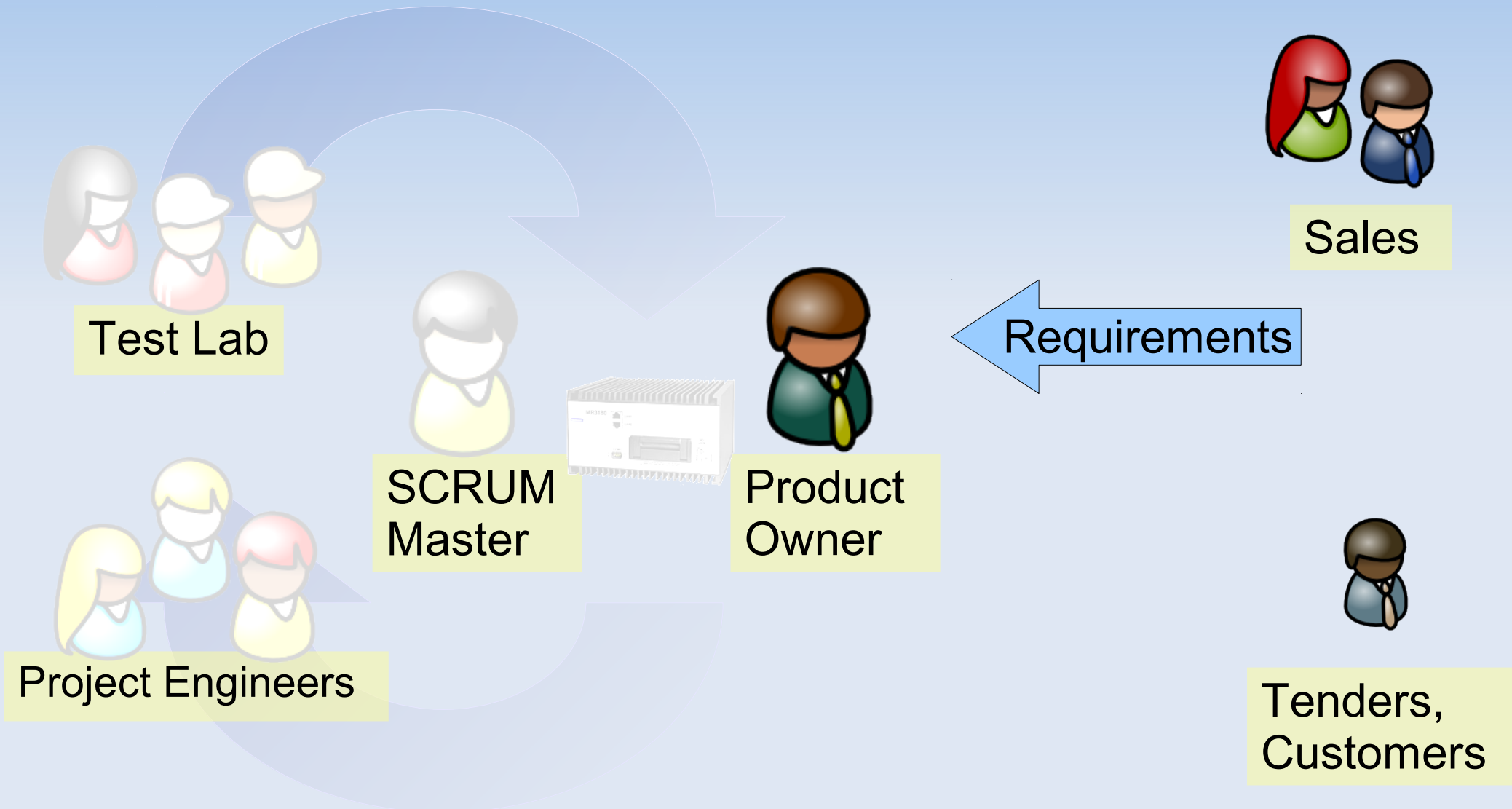
Lightweight Design

*Design Specs*

*(where necessary)*



# Roles and Parties revisited



# Roles and Parties revisited



Test Lab



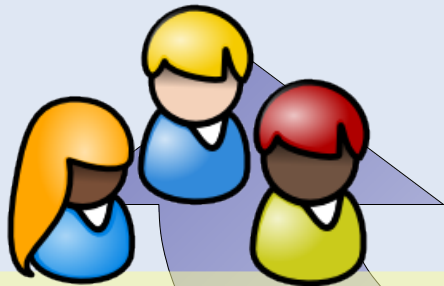
Sales



SCRUM Master



Product Owner

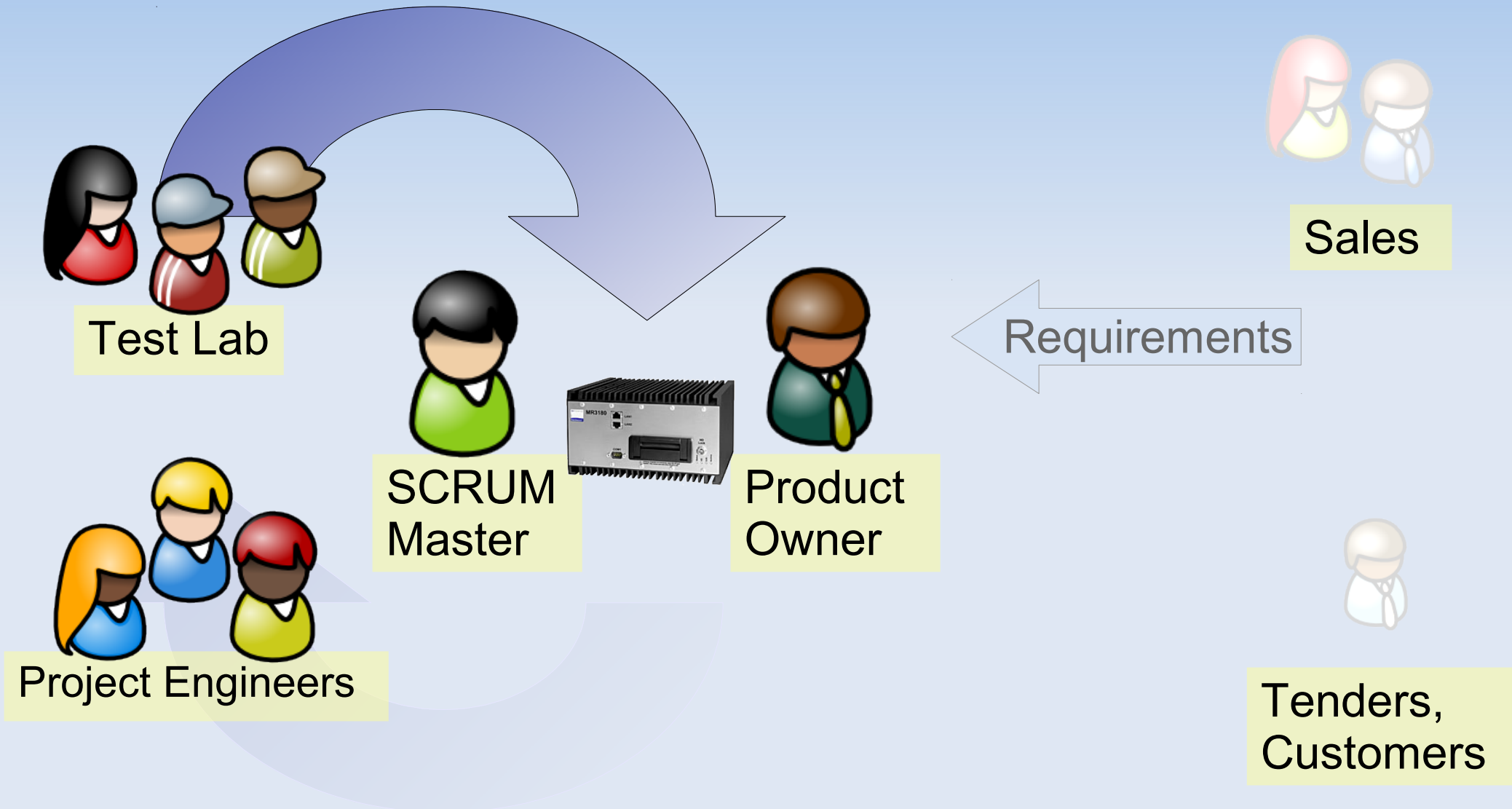


Project Engineers

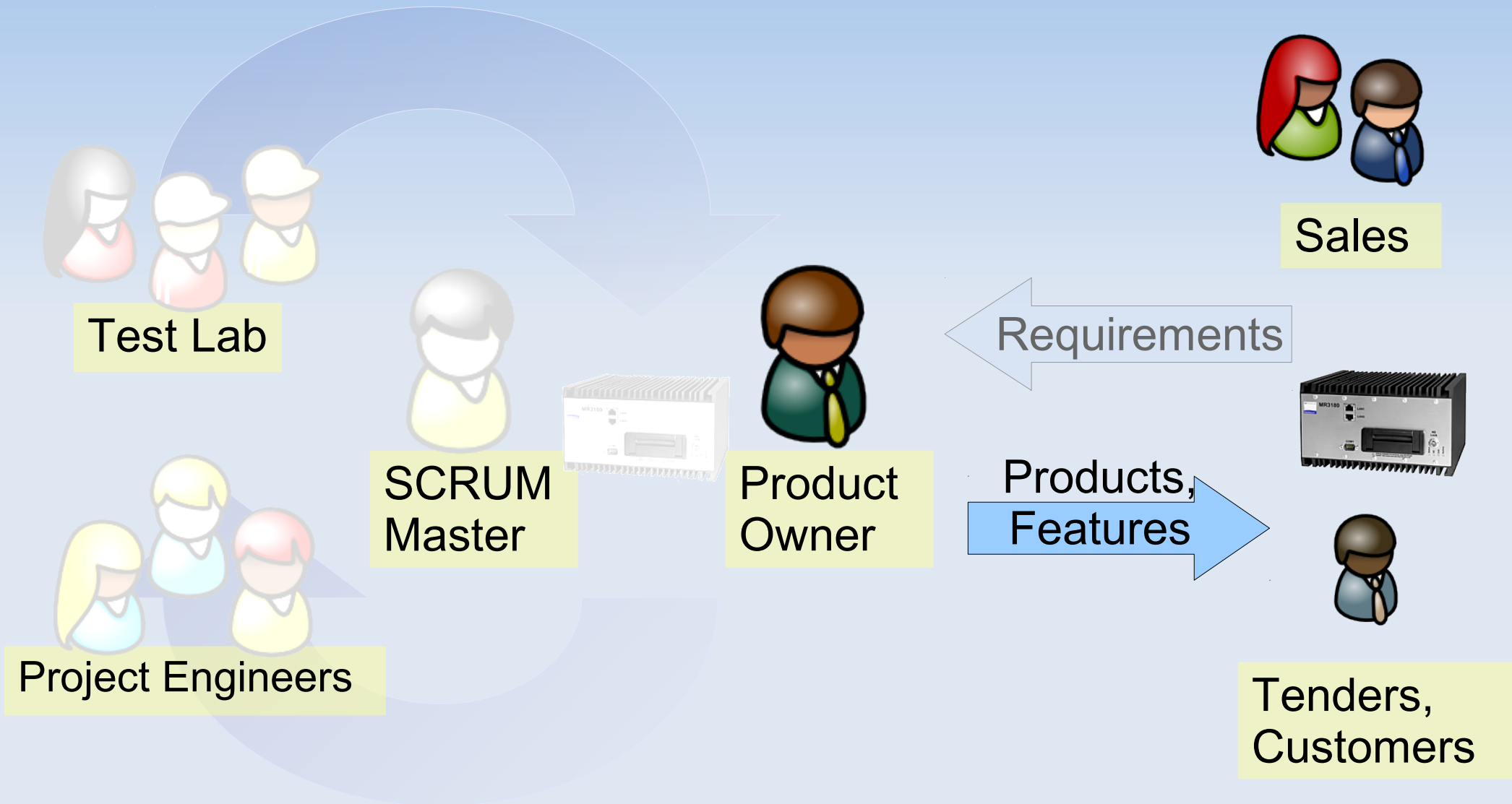


Tenders,  
Customers

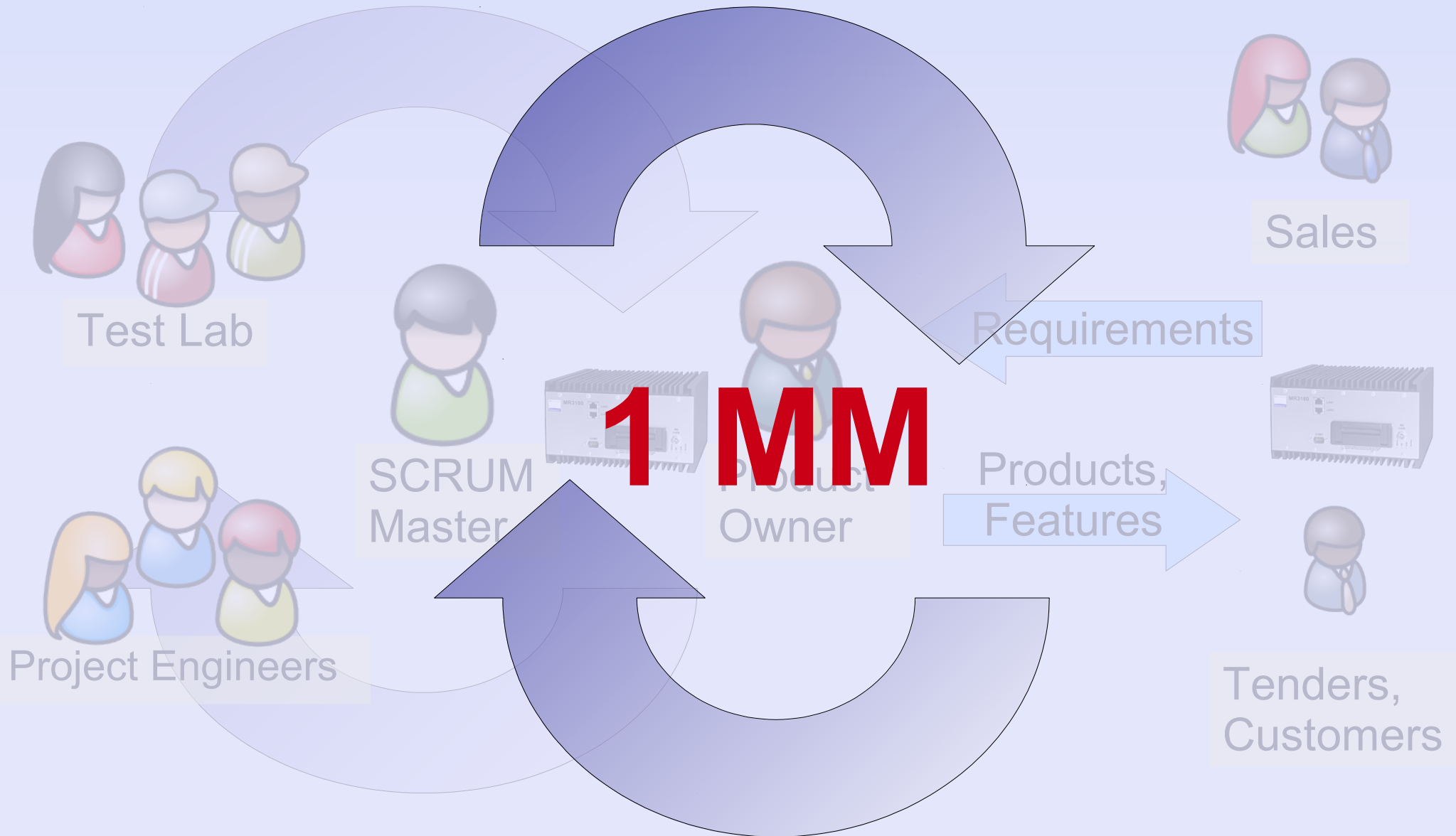
# Roles and Parties revisited



# Roles and Parties revisited



# Roles and Parties revisited



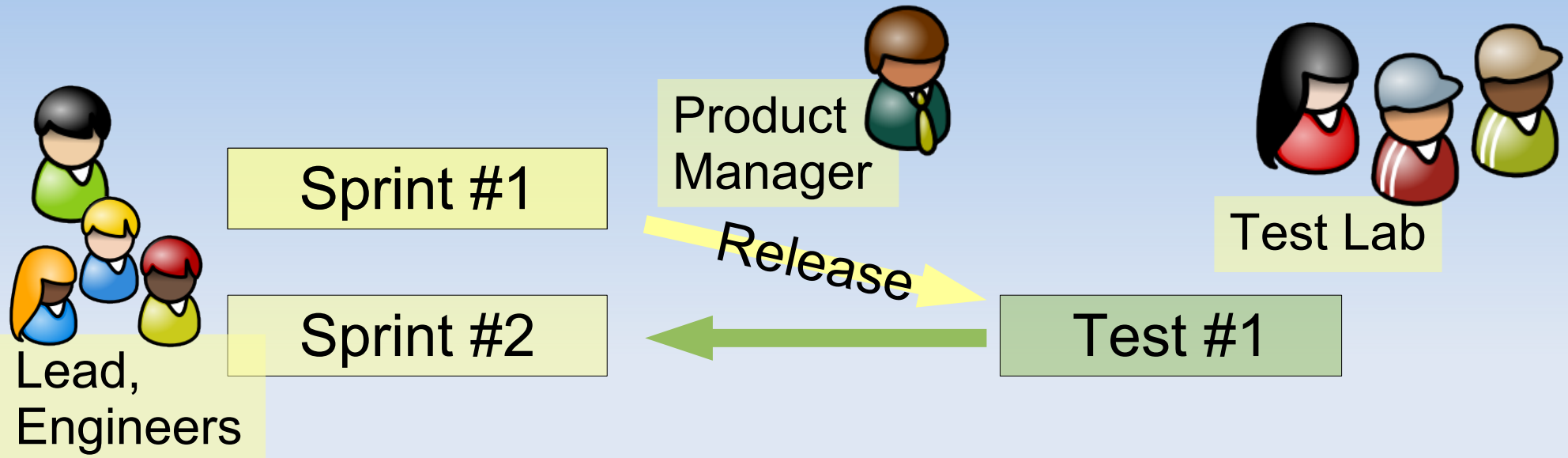
# Implementation / Verification rvst.



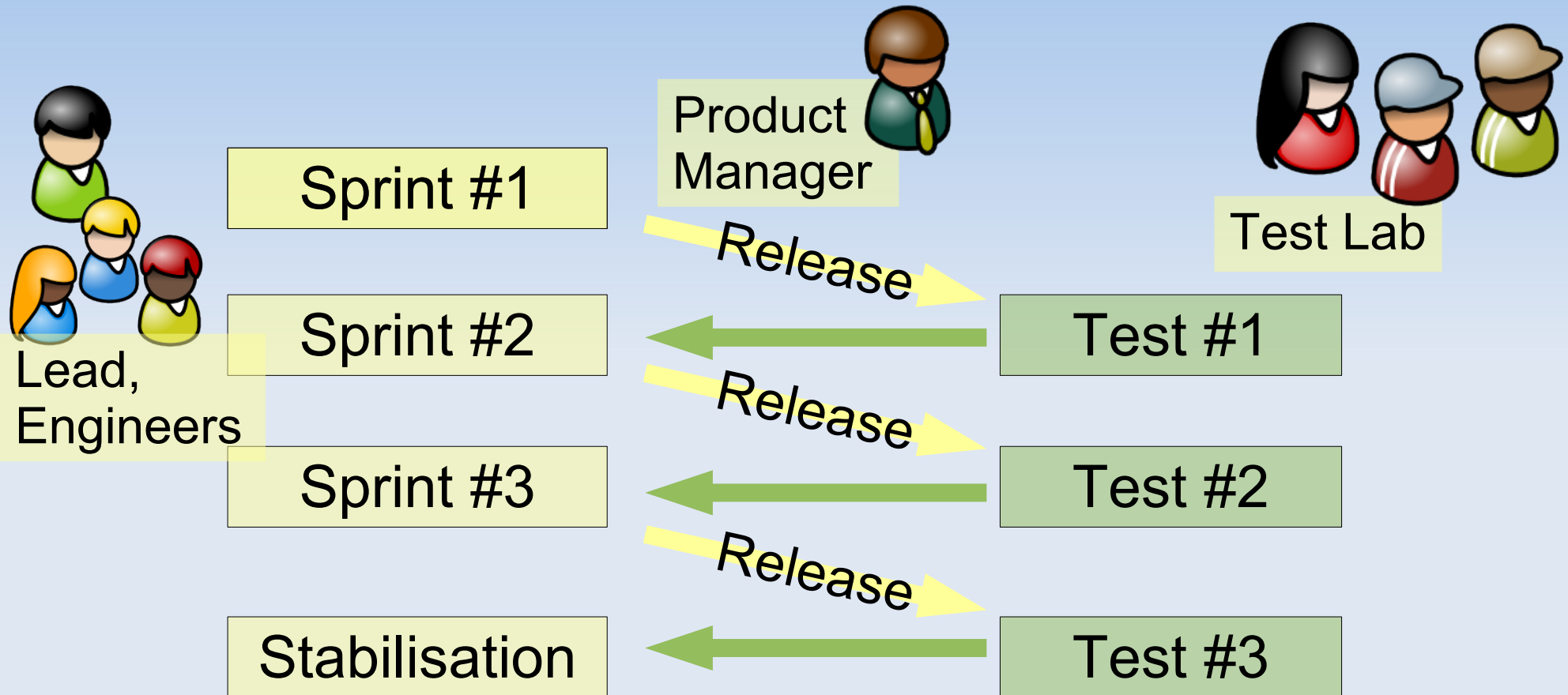
Sprint #1



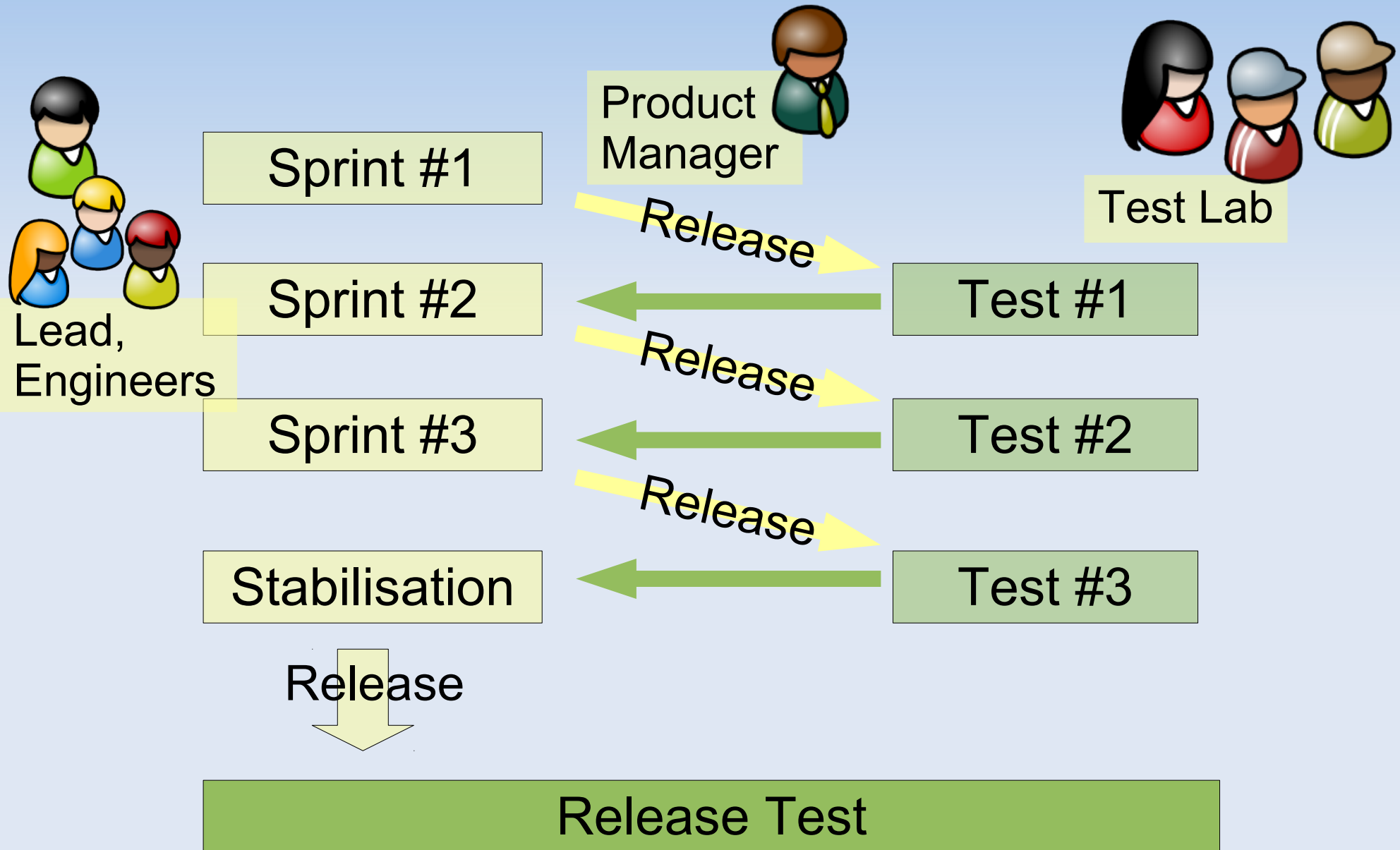
# Implementation / Verification rvst.



# Implementation / Verification rvst.



# Implementation / Verification rvst.



# Lessons learned

## SCRUM is **hard**

Programming paradigms change (TDD, RERO)

Management paradigms change

## Early feedback is (your) money

Find, fix and verify bugs early

Communicate requirement changes ASAP

# Lessons learned

Waterfall / Agile try to solve different problems

Both are appropriate for their problem class

Don't decide for one, take the best of both

Eliminate deficiencies of both methods along the way

”If you think Agile will solve your problems, then you don't understand Agile, and you don't understand your problems.”

# Q&A // Commitments

”The changes introduced enable coordinated response to customer requirements and enhance strategic marketing flexibility. We received Declarations of Satisfaction from several customers.”

**Peter Schüßler, Product Manager**

”Due to the very positive experience we made by introducing Sprints into the Waterfall process we will continue to apply this novel model to this year's upcoming projects.”

**Frank Bauernöppel, Head of Product Development**

# Q&A // Commitments cont'd.

”From a Controller's perspective I still mourn the days where a project had time, cost and features set fix before development even started. Calculating and reporting progress was an easy and controller-friendly task back then.”

However, introducing agile principles into DResearch processes enabled us to process projects wich featured rapidly changing requirements, varying feature priorities and time critical release constraints with much more customer satisfaction than the Waterfall method could provide.

As for our motto being customer satisfaction, not controller's satisfaction, I cannot help but support the future use of Agile Principles at DResearch ;-))”

Susanne Bärsch, Head of Testing / Controlling

**Thanks!**

This presentation is released under the terms of the GNU Free Documentation License, version 1.2, or later, as described in <http://www.gnu.org/copyleft/fdl.html> .

Get the sources from

[https://www.thilo-fromm.de/~t-lo/slides/SCRUM\\_Waterfall/](https://www.thilo-fromm.de/~t-lo/slides/SCRUM_Waterfall/)

# List of Sources

Title image: Purakaunui.jpg by James Dignan, © GNU FDL, from  
<http://de.wikipedia.org/wiki/Wasserfall>