Project Management with TRAC

By Glib Kutepov (IESE)

© Fraunhofer IESE

Motivation

- Project Managers will have to manage the project
- It will be not that simple
- You have a complex project setting
    - You use Iterative approach
    - Iterations are rather short
    - Different Roles
    - Large Group of Students
- *How will you manage people?*
- *How will you assign tasks?*
- *How will you know if the project is going good?*
- *How will you steer the project?*
- Management and communication effort will be pretty big
- You can lower the effort via **tool support**

© Fraunhofer IESE

Motivation

- What do we mean by **tool support**??
- **You need to know:**
  - **What** do you want to support with that tool?
  - Which **tool** supports that the best?
  - How do you **use** that tool?

© Fraunhofer IESE

Outline

1. **What** to support?
2. **Which** tool to use?
3. **How** to use this tool best?

© Fraunhofer IESE

Outline

1. **What** to support?
2. **Which** tool to use?
3. **How** to you the tool best?

© Fraunhofer IESE

What to support?

- You need to support **project**
- Project runs according to the **process model**
- So, you need **process model support**
- What is **SE** process model?
    - The set of activities and the sequence in which they have to be carried out
- So what is **your** SE process model?

© Fraunhofer IESE

What is your process model?

- What are the activities?
  - The set is strict (Requirements, Architecture, Implementation, Quality Assurance)
- Is that waterfall?
  - You use iterations
  - You have people working in parallel
  - Working times vary
- Is that Agile Process (e.g. Scrum)?
  - You don’t have Scrum Roles (Product Owner, Scrum Master)
  - You don’t use Scrum terminology (Sprint, Backlog, Burndown)
- **This is an iteration based process with agile flavor**
  - You are free to choose the best of two worlds

© Fraunhofer IESE

What is a process in general?

- Set of **activities**
- Carried out by **team members**
- In a specific **order**
- whithin specific **timeframe**
- Aimed at producing **deliverables**
- Example: **John** has to complete the **Requirements Document** by **Thursday the 29th**

© Fraunhofer IESE

What is the responsibility of the Project Manager during the process?

- Create and schedule the set of **activities**
- Make sure **team members** aware and agree
- Make sure the appropriate **deliverables** are delivered on **time**

- **Activities** – RE, Architecture, Development, QA (different levels of detail)
- **Team members**
- **Deliverables** – documents, diagrams, minutes, code etc.

You probably want your responsibilities to be supported by the tool?

© Fraunhofer IESE

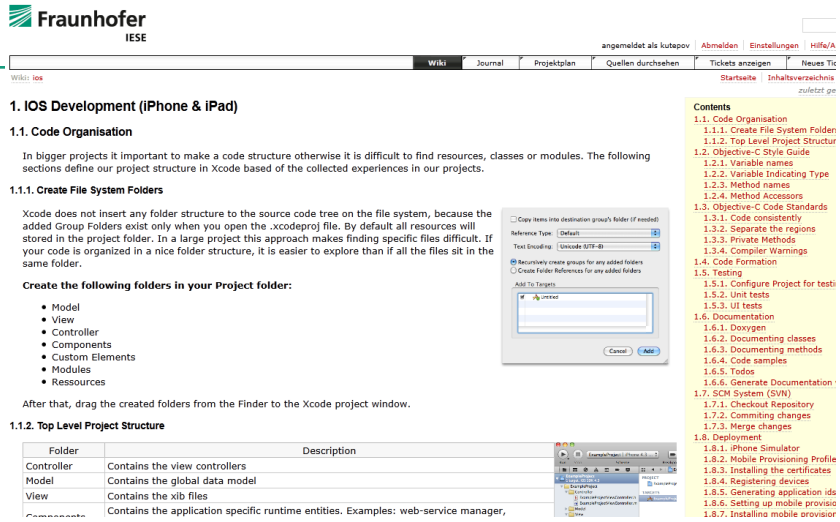
Outline

1. What to support?
2. **Which** tool to use?
3. How to use this tool best?

© Fraunhofer IESE

TRAC?

1. Web based software project management tool
2. User management (manage your team)
3. Ticketing System (manage your activities/tasks)
4. Project Planning (manage the order of activities)
5. Interface to versioning systems (manage your deliverables)
6. Wiki (manage your deliverables)
7. Timeline (control your project)
8. Plugins
    - Time Tracking
    - Ticket visualization
    - …



© Fraunhofer IESE

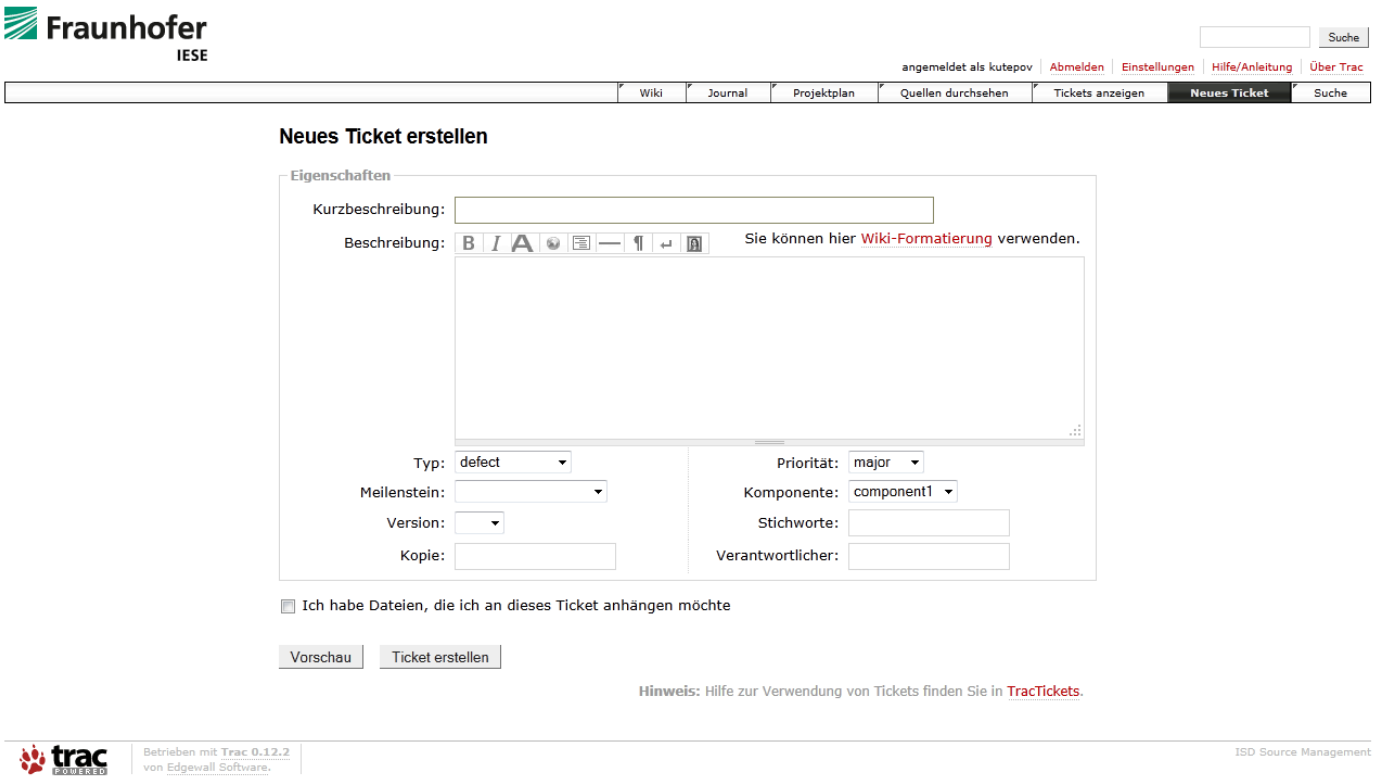
Fraunhofer IESE

TRAC user management

- Users will be created automatically by IESE

© Fraunhofer IESE

TRAC ticketing system 1



© Fraunhofer IESE

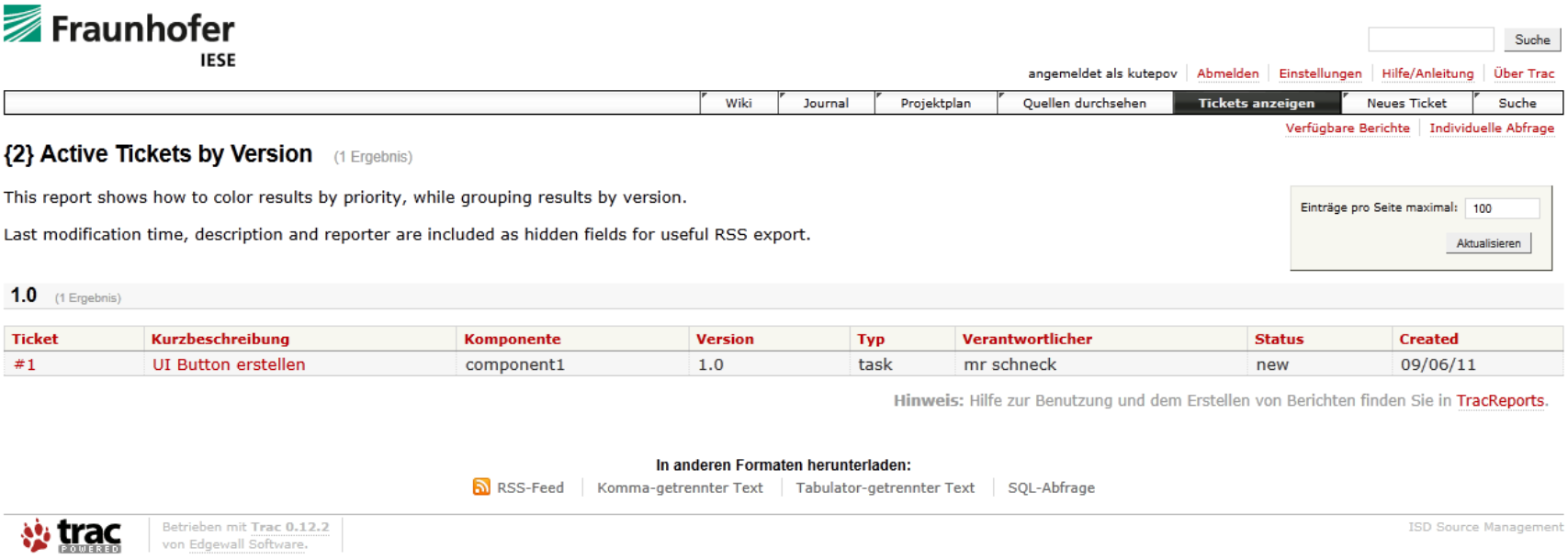
Fraunhofer
IESE

TRAC tickets

1. Tickets have:
    - **Reporter** — The author of the ticket.
    - **Type** — Task, Defect, Change Request
    - **Component** — The project module or subsystem this ticket concerns.
    - **Version** — Version of the project that this ticket pertains to.
    - **Priority** — The importance of this issue
    - **Milestone** — milestone which this ticket belongs to
    - **Assigned to/Owner** — Responsible person
    - **Resolution** —Fixed, invalid, wontfix, duplicate, worksforme.
    - **Status** —New, assigned, closed, reopened.
    - **Description** — clear

© Fraunhofer IESE

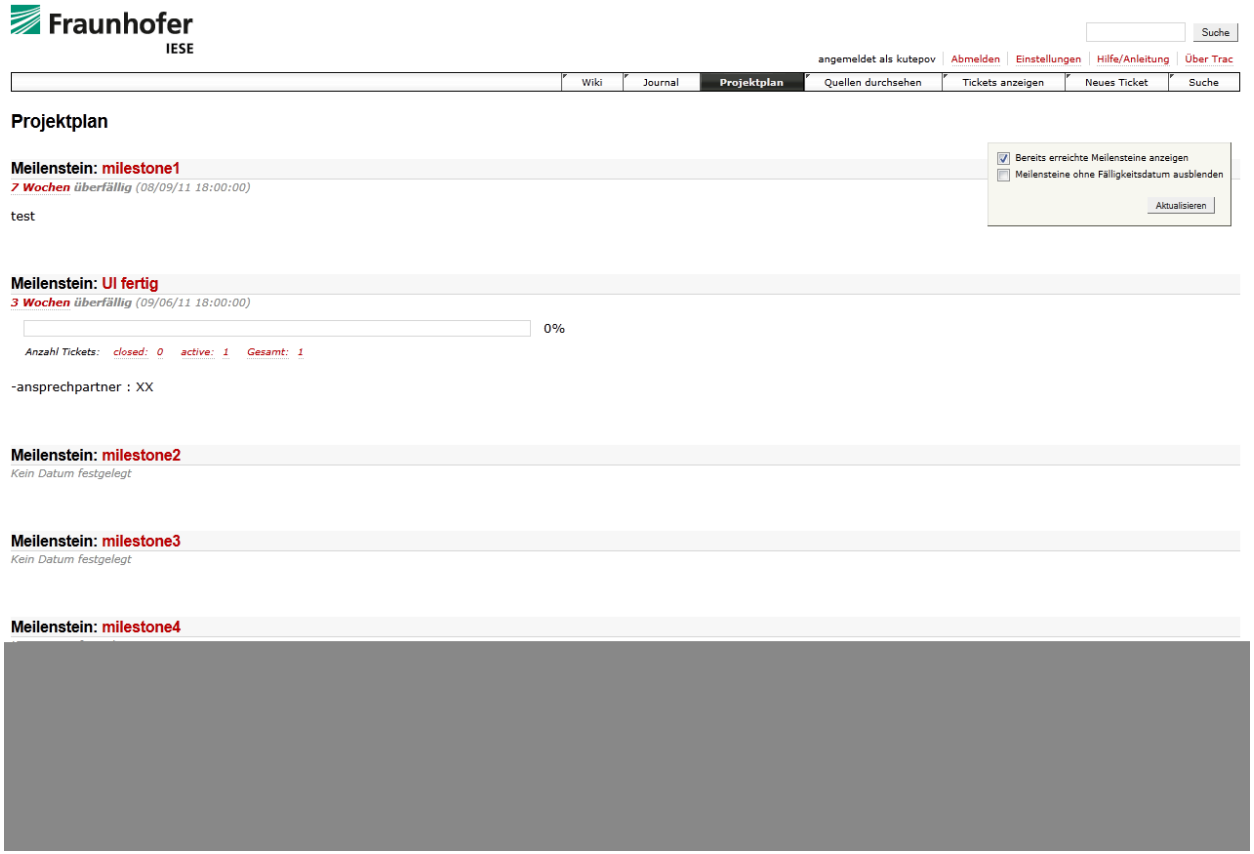
TRAC ticketing system 2



© Fraunhofer IESE

TRAC project planning

1. **Milestone** is the **set of the tickets** which need to be done within a certain **timeframe**



© Fraunhofer IESE

Fraunhofer IESE

TRAC wiki

Fraunhofer IESE

angemeldet als kutepov | Abmelden | Einstellungen | Hilfe/A

Wiki | Journal | Projektplan | Quellen durchsehen | Tickets anzeigen | Neues Tic

Wiki: ios

Startseite | Inhaltsverzeichnis

zuletzt ge

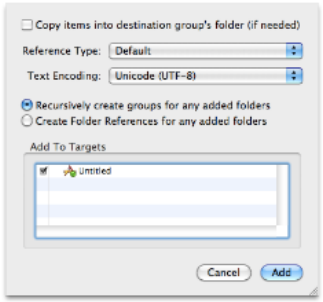
1. IOS Development (iPhone & iPad)

1.1. Code Organisation

In bigger projects it important to make a code structure otherwise it is difficult to find resources, classes or modules. The following sections define our project structure in Xcode based of the collected experiences in our projects.

1.1.1. Create File System Folders

Xcode does not insert any folder structure to the source code tree on the file system, because the added Group Folders exist only when you open the .xcodeproj file. By default all resources will stored in the project folder. In a large project this approach makes finding specific files difficult. If your code is organized in a nice folder structure, it is easier to explore than if all the files sit in the same folder.



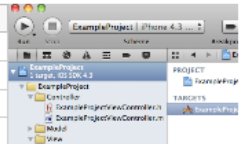
Create the following folders in your Project folder:

- Model
- View
- Controller
- Components
- Custom Elements
- Modules
- Ressources

After that, drag the created folders from the Finder to the Xcode project window.

1.1.2. Top Level Project Structure

Folder	Description
Controller	Contains the view controllers
Model	Contains the global data model
View	Contains the xib files
Components	Contains the application specific runtime entities. Examples: web-service manager,



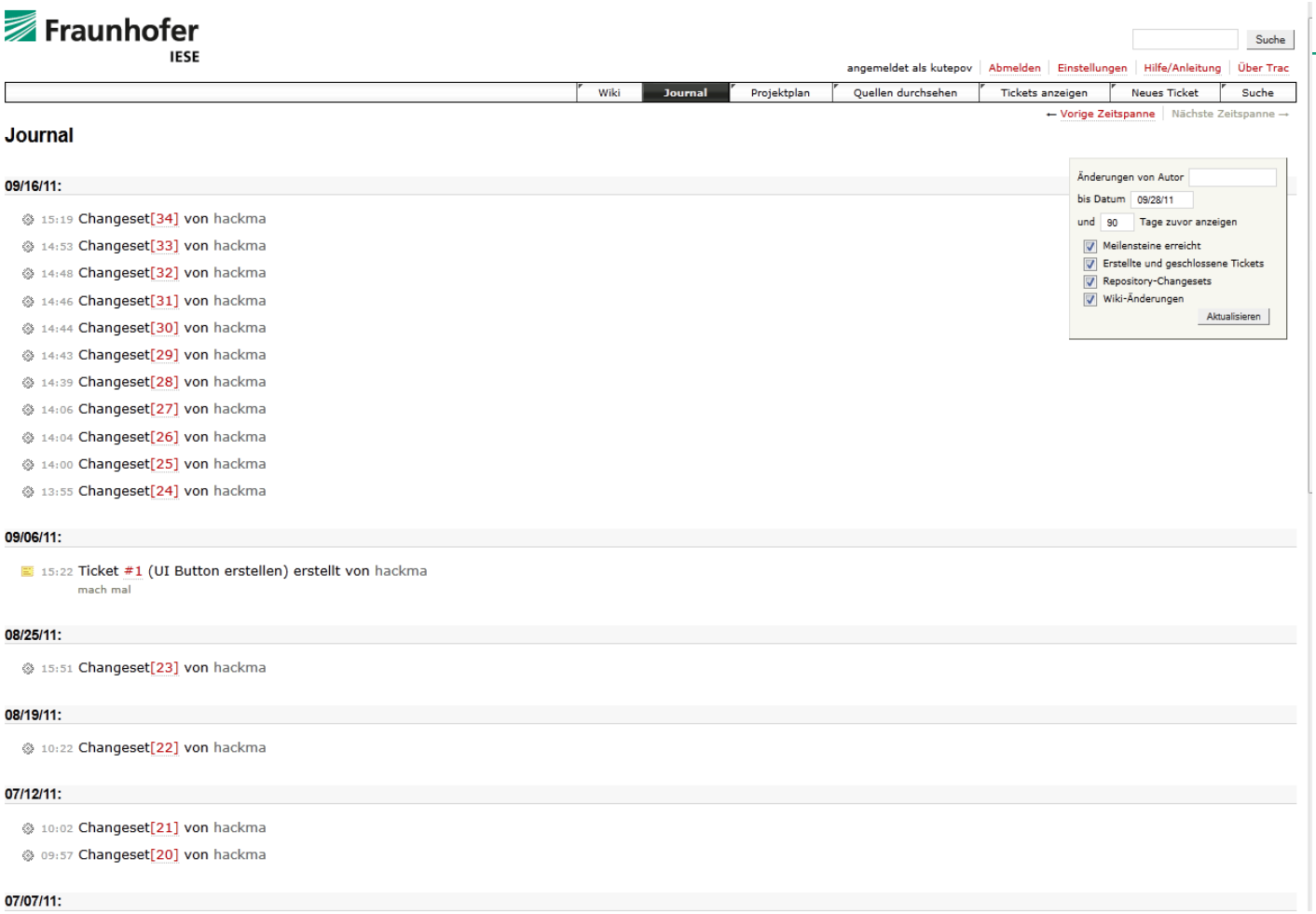
Contents

1.1. Code Organisation
1.1.1. Create File System Folder
1.1.2. Top Level Project Structur
1.2. Objective-C Style Guide
1.2.1. Variable names
1.2.2. Variable Indicating Type
1.2.3. Method names
1.2.4. Method Accessors
1.3. Objective-C Code Standards
1.3.1. Code consistently
1.3.2. Separate the regions
1.3.3. Private Methods
1.3.4. Compiler Warnings
1.4. Code Formation
1.5. Testing
1.5.1. Configure Project for testi
1.5.2. Unit tests
1.5.3. UI tests
1.6. Documentation
1.6.1. Doxygen
1.6.2. Documenting classes
1.6.3. Documenting methods
1.6.4. Code samples
1.6.5. Todos
1.6.6. Generate Documentation
1.7. SCM System (SVN)
1.7.1. Checkout Repository
1.7.2. Commiting changes
1.7.3. Merge changes
1.8. Deployment
1.8.1. iPhone Simulator
1.8.2. Mobile Provisioning Profile
1.8.3. Installing the certificates
1.8.4. Registering devices
1.8.5. Generating application ids
1.8.6. Setting up mobile provisio
1.8.7. Installing mobile provisior

© Fraunhofer IESE

Fraunhofer IESE

TRAC timeline



© Fraunhofer IESE

Looks good?

- Not quite…
- Have you seen **milestones** in our process?
- Have you seen **activities** (RE, Architecture etc.) and **iterations** in TRAC?
- We have to match these things!
- **TRAC is not process specific – it is just a framework which we need instantiate for our process**

© Fraunhofer IESE

Outline

1. What to support?
2. Which tool to use?
3. **How** to use this tool best?

© Fraunhofer IESE

Instantiate Trac for our Process? **Problem.**

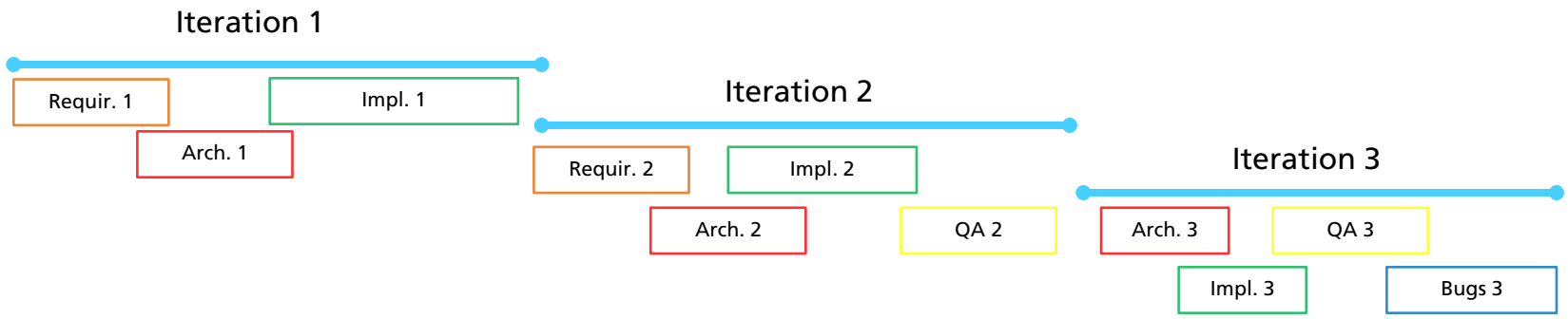
1. **Our Dimensions**
    - Products/Activites (Requirements, UI, Architecture, Implementation, QA)
    - Time (3 Iterations/Releases)
    - We need to do something for each product in each iteration
2. **Trac Dimensions**
    - Time and Products (Milestones)
    - Tasks related to products (Tickets)

How to express the first with the second?

© Fraunhofer IESE

Instantiate Trac for our Process? **Solution.**

1. Slice the products in time
    - Requirements 1 (Iteration 1), Requirements 2 (Iteration 2)
    - QA 2 (Iteration 2), QA 3 (Iteration 3)
2. Create a milestone for each timesliced product
3. Completion of all the milestones related to the iteration denotes the completion of the iteration
4. Milestones consist of tasks (trac: tickets)
    - E.g.: Requirements 1: “Create personas”, “Complete Use Cases 1 - 10” etc.



© Fraunhofer IESE

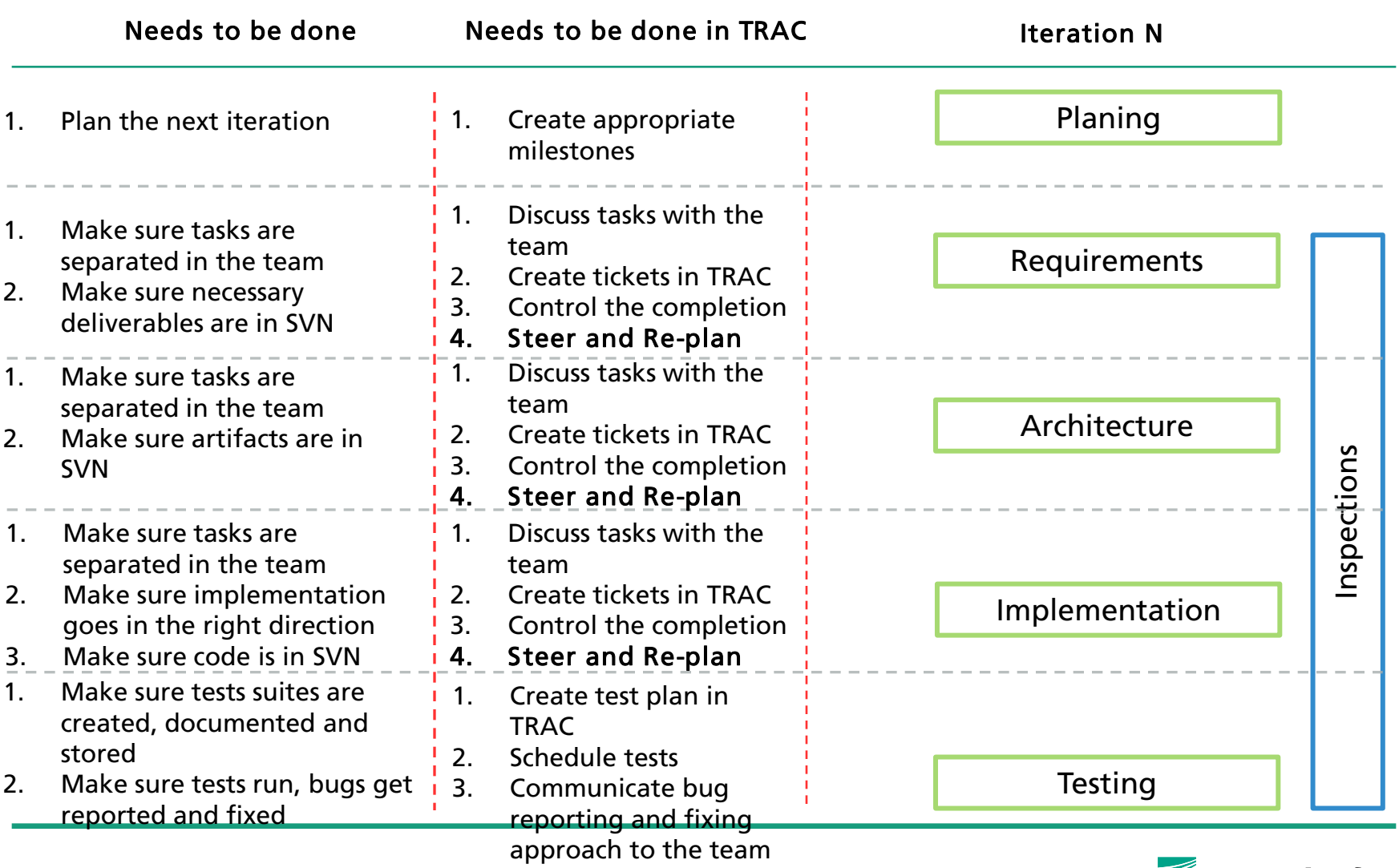
Fraunhofer IESE

Using tickets for different activities

1. Requirements
   - Simple tasks: e.g. “Create traceability matrix”, “Add minutes to SVN”
2. Architecture
   - E.g. “Create C&C View”, “Finish Architectural Scenarios”
3. Development
   - Development tickets related to particular architectural components
4. QA
   - Tickets as inspection tasks
   - Bugs (use tickets as issue trackers)
   - Test planning (use plugin for test planning)

© Fraunhofer IESE

Tasks of a Project Manager



© Fraunhofer IESE

Fraunhofer IESE

Re-planing?

- Things will most certainly go wrong
  - Don’t be afraid!
- Steer the process
- Discuss with the team
- Rearrange tickets and milestones
- Continue!

© Fraunhofer IESE

Some recommendations

1. Be a Team
2. Understand the needs of each other
    - Project Managers need Others
    - Others need Project Managers
3. Communicate
    - Share good and bad
    - Don’t be embarrassed to ask for help
4. Project managers
    - Use your **soft skills**
    - **Remember: no process, plan or tool will substitute the person!**

© Fraunhofer IESE

Thank you for your attention!

© Fraunhofer IESE