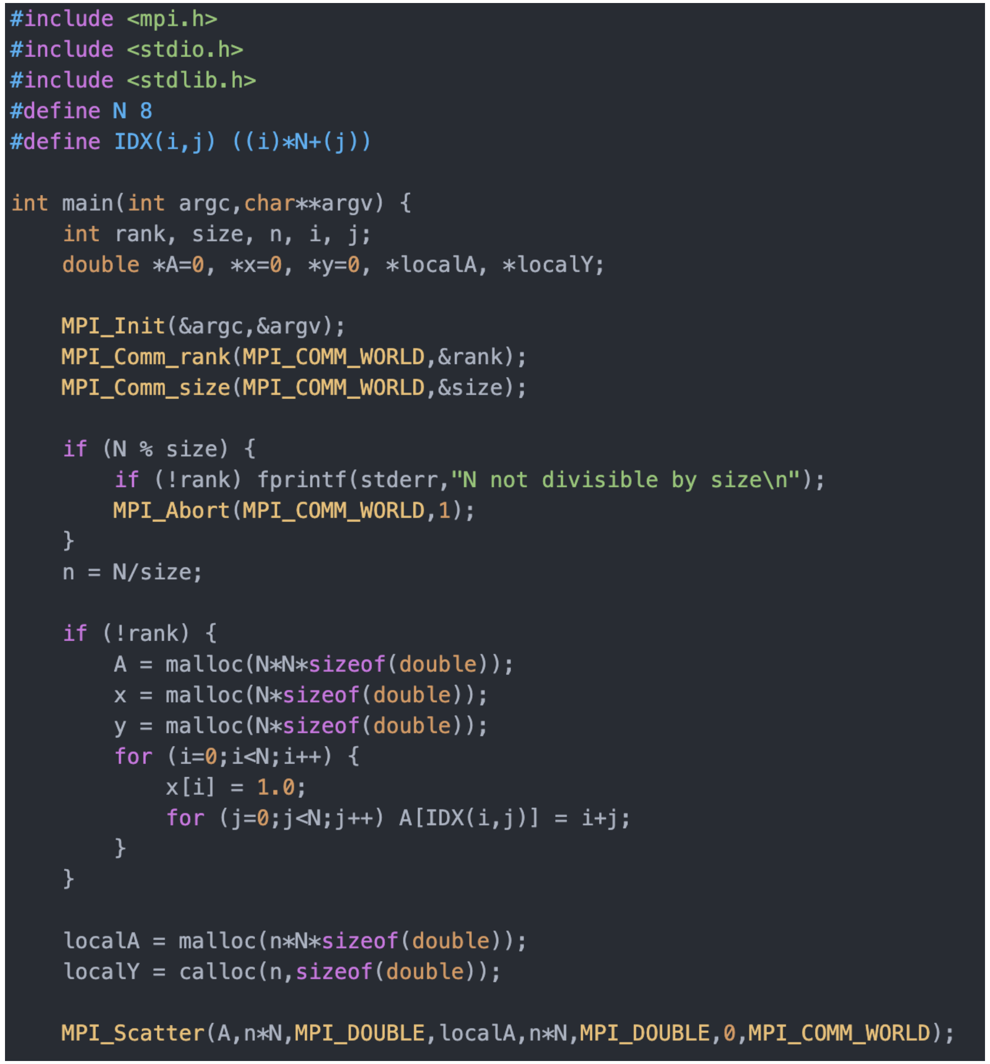
The Jülich Research Software Engineering Community of Practice – JuRSE

24th September 2025 I Robert Speck for the JuRSE team

Member of the Helmholtz Association

Domain scientists/HPC experts write great code!



```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#define N 8
#define IDX(i,j) ((i)*N+(j))

int main(int argc,char**argv) {
    int rank, size, n, i, j;
    double *A=0, *x=0, *y=0, *localA, *localY;

    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&size);

    if (N % size) {
        if (!rank) fprintf(stderr,"N not divisible by size\n");
        MPI_Abort(MPI_COMM_WORLD,1);
    }
    n = N/size;

    if (!rank) {
        A = malloc(N*N*sizeof(double));
        x = malloc(N*sizeof(double));
        y = malloc(N*sizeof(double));
        for (i=0;i<N;i++) {
            x[i] = 1.0;
            for (j=0;j<N;j++) A[IDX(i,j)] = i+j;
        }
    }

    localA = malloc(n*N*sizeof(double));
    localY = calloc(n,sizeof(double));

    MPI_Scatter(A,n*N,MPI_DOUBLE,localA,n*N,MPI_DOUBLE,0,MPI_COMM_WORLD);
```

Heard that one before? How about:

🗣 "I'm an expert, my code is great!"

🗣 "Meh, nobody looks at my code anyway.."

🗣 "I (and only I) know my code best!"

🗣 "Can you make this faster, please?"

Assumption: HPC users = code professionals, i.e.

- The code is readable and understandable
- The code is documented properly
- It comes with a readme and installation guides
- Examples show how to use it
- It has **tests to show that it works** properly
- It is citable

Observation: often not true

Are you working with research software? (Spoiler: yes)

Examples of research software

- An application software, serial, parallel, GPU, CPU, big machine, small machine
- A software other people use for their scientific work: a benchmarking tool, a profiler, a plotting script

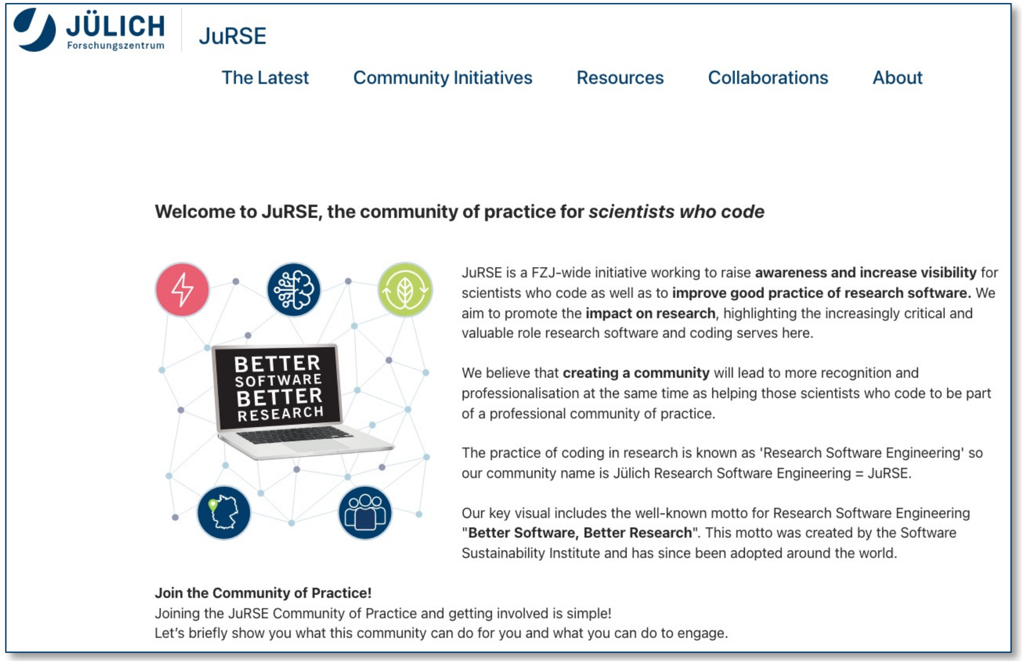
Gathering under the term “Research Software Engineering” follows two goals

1. Enable scientists who code, students, research software engineers, HPC users, domain experts, you, to write good research software
2. Make your work visible, create awareness for your work

Getting there: an RSE community of practice

- Provide a platform for people to exchange thoughts on research software, help them find their tribe
- Make software count as research output, make software work count as a valuable part of science

JuRSE – Jülich RSE Community of Practice



Welcome to JuRSE, the community of practice for *scientists who code*

JuRSE is a FZJ-wide initiative working to raise **awareness and increase visibility** for scientists who code as well as to **improve good practice of research software.** We aim to promote the **impact on research**, highlighting the increasingly critical and valuable role research software and coding serves here.

We believe that **creating a community** will lead to more recognition and professionalisation at the same time as helping those scientists who code to be part of a professional community of practice.

The practice of coding in research is known as 'Research Software Engineering' so our community name is Jülich Research Software Engineering = JuRSE.

Our key visual includes the well-known motto for Research Software Engineering "**Better Software, Better Research**". This motto was created by the Software Sustainability Institute and has since been adopted around the world.

Join the Community of Practice!
Joining the JuRSE Community of Practice and getting involved is simple!
Let's briefly show you what this community can do for you and what you can do to engage.

https://go.fzj.de/JuRSE

- **Increase good practice, but also visibility and awareness**
- **Encourage adoption of the software and publication guidelines**

JuRSE – The Team

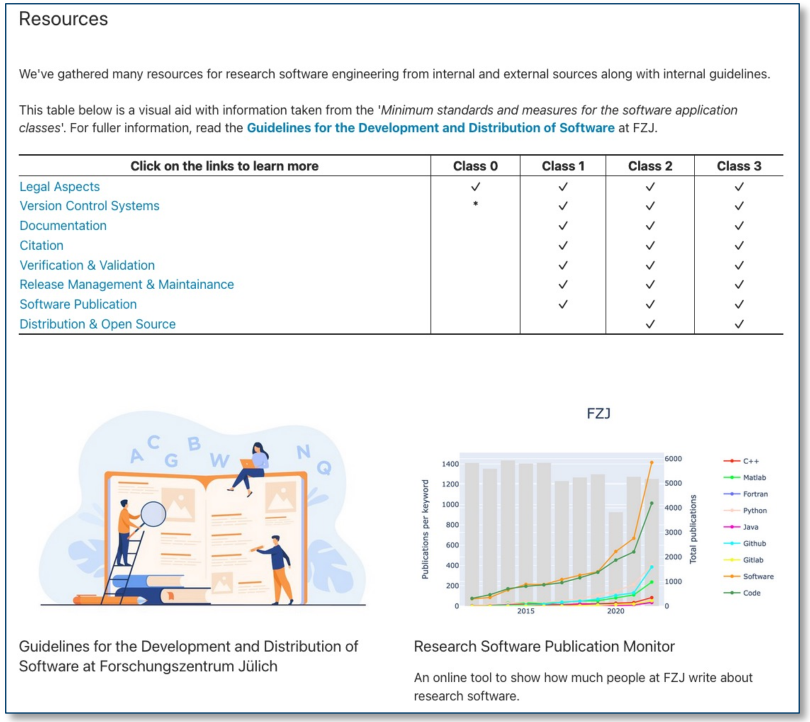
Currently:

- Dirk Brömmel
- Jakob Fritz (partially)
- Robert Speck - team lead
- Dorothee Stängle
- Claire Wyatt
- Plus: representatives from the library, legal, and more

- **Increase good practice, but also visibility and awareness**
- **Encourage adoption of the software and publication guidelines**

JuRSE – Supporting the FZJ Software Guidelines

GUIDELINES FOR THE DEVELOPMENT AND DISTRIBUTION OF SOFTWARE AT FORSCHUNGSZENTRUM JÜLICH



Resources

We've gathered many resources for research software engineering from internal and external sources along with internal guidelines.

This table below is a visual aid with information taken from the '*Minimum standards and measures for the software application classes*'. For fuller information, read the **Guidelines for the Development and Distribution of Software** at FZJ.

Click on the links to learn more	Class 0	Class 1	Class 2	Class 3
Legal Aspects	✓	✓	✓	✓
Version Control Systems	*	✓	✓	✓
Documentation		✓	✓	✓
Citation		✓	✓	✓
Verification & Validation		✓	✓	✓
Release Management & Maintainance		✓	✓	✓
Software Publication		✓	✓	✓
Distribution & Open Source			✓	✓

Guidelines for the Development and Distribution of Software at Forschungszentrum Jülich

Research Software Publication Monitor

An online tool to show how much people at FZJ write about research software.

- Increase good practice, but also visibility and awareness
- Encourage adoption of the software and publication guidelines

JuRSE – Community and visibility activities

Join the community platform (online)

Meet other RSEs at FZJ and Germany!

Join us at JuRSE Open Hours

In-person and online, we're here to help!

JuRSE Travel Grants

Go to an RSE Conference on us!

JuRSE Code of the Month

A code in the spotlight

JuRSE Newsletters

Read about RSE News, events, blogs and podcasts

HIRSE

- **Increase good practice, but also visibility and awareness**
- **Encourage adoption of the software and publication guidelines**

Find out more!

https://go.fzj.de/JuRSE