Fork-based (or branch-based) development is a lightweight mechanism that allows developers to collaborate with or without explicit coordination. Recent advances in distributed version control systems (e.g., ‘git clone’) and social coding platforms have made fork-based development relatively easy and popular by providing support for tracking changes across multiple forks with a common vocabulary and mechanism for integrating changes back. However, fork-based development has well-known downsides. When developers each create their own fork and develop independently, their contributions are usually not easily visible to others, unless they make an active attempt to merging their changes back into the original project. When the number of forks grows, it becomes very difficult to keep track of decentralized development activity in many forks. The key problem is that it is difficult to maintain an overview of what happens in individual forks and thus of the project’s scope and direction. Furthermore, the problem of lacking an overview of forks can lead to several additional problems and inefficient practices: lost contributions, redundant development, fragmented communities, and so on.

Facing the problems mentioned above, we would like to alleviate these inefficiencies. We developed two complementary strategies: Identifying existing best practices and designing new interventions. First, during the process of sampling 1355 GitHub projects and quantifying the inefficiencies, also by opportunistically reaching out to developers who have used forks, we recognized that there are differences among projects. Therefore, we would like to identify existing best practices and suggesting evidence-based interventions for projects that are inefficient. Second, awareness solutions increase the transparency in collaborative software development, there is a lot of information that is publicly available but not easily accessible. Thus, we would like to build tools that could improve the awareness and transparency of a community using fork-based development, and help developers to detect redundant development to reduce developers’ unnecessary effort. To evaluate the effectiveness and usefulness of this work, we propose to conduct both quantitative and qualitative studies.