

**MASTER'S THESIS** Investigating the impact of code sharing and how to manage it

**STUDENTS** André Alm & Daniel Dornlöv

**SUPERVISORS** Lars Bendix (LTH) & Albert Rigo (Praqma)

**EXAMINER** Ulf Asklund (LTH)

# How to organize and manage shared code

---

POPULAR SCIENCE SUMMARY **André Alm & Daniel Dornlöv**

---

In today's development climate it is becoming more and more popular to share code in order to save on time, rework and costs. This work looks at the motivations and problems that exists for companies in a shared code context, as a part of this a case study at a company is made to get an industry perspective and to give recommendations on how they can work with shared code.

Many companies working with software development today would like to achieve a higher speed of the development process and avoid having to write the same code again. To try and keep the costs at a reasonable level, a rising trend is to share code between projects, teams or developers. The shared code might be source code, binaries, or libraries.

This thesis focuses on what drives a company to start using shared code and also on what can be done, both to try to solve the problems caused by code sharing, but also to try and see how shared code can be supported so that it becomes as effective and efficient as possible. To get an understanding of what different drivers and problems there are with code sharing interviews were conducted with both developers and persons working in a more overarching role between projects and teams. In addition to the interviews a literature study was conducted and the findings from both of these were weighed together to form a requirements specification on what is needed to share code and avoid the problems identified during the interviews and literature study.

The requirements were used as a guide to find solutions to the problems with shared code. It be-

came apparent that there is no silver bullet that can solve all of the problems, instead the trick is in combining solutions to form a package of tools, principles and workflows that together mitigate or lessens the problems. This approach led to a matrix in which it is possible to find a solution based on at what phase in the development a problem occurs and on what type of solution it is. This matrix was then used to make a case study at a company and give recommendations on how they could work with shared code.

For the case study company, two different recommendations were made because mainly two different user scenarios were identified, and these require different measures. The first perspective dealt with sharing code between product teams involving many people and a wide spectrum of products while the second is related to sharing on a smaller scale between a few developers, often in experimental or exploratory purposes. A difficulty with the recommendations is that it become clear that a solution that works now will not necessarily work in a few years (or less) time. This prompted the recommendations to try and look a bit forward and take into consideration the switching of solution when that time comes.