Software systems are today bigger, more complex and of higher importance for products and services than a decade before. At the same time changes are required many more frequently and of a larger size. Furthermore, they have to be implemented faster. Additionally, the software must achieve a higher life span, particularly because of the cost of its development. In the past, Object-Oriented Programming and Reuse techniques did not provide the expected success. The introduction of software product lines respectively system families makes possible it to reach a degree of prefabrication similar to the one of serial production. At the same time they facilitate the delivery of product variants with a short time to market.

In this work methods of the methods of domain analysis are integrated with Reuse approaches and techniques of Generative Programming, and a methodology for product line development is presented. Feature models are used as means expressing variability and product configurations, so that the prefabrication be planned and the production of customer-specific products can be controlled. By enforcing the formalization in terms of syntax and semantics, feature models are made accessible to tools and automation. Object-oriented design models and architecture are separated into fine-granular components in such a way that new products can easily be developed as combinations of those components. The implementation of such products is automated by the composition of source code components. The composition of object models separated similarly enables a uninterrupted automation for the product development, which is controlled by a customer by means of a feature selection. To facilitate such a composition, the Hyperspace approach is applied to UML to Hyper/UML, which makes possible a feature-driven separation and composition of object models. In this way slim products can be developed, containing only the actually needed functionality. For the evolution of product lines and for the integration of existing solutions and components into the evolution, Reverse Engineering and Refactoring techniques are integrated. Requirements, models and implementation are connected by Traceability links to perform changes consistently. As a consequence, the loss of architectural quality - so-called Architectural Decay - can be avoided during the iterative development process. Measures for the improvement of the project and quality management are regarded briefly, as far as they are of importance for the effectiveness of the developed methods. The applicability and suitability of the results of the work were examined in several industrial projects.