**PLUM (Product Line Unified Modeller)**

*Eclipse based Variability Management Tool*

**The context**
Companies focused in a specific market segment deliver products that share many features in common. The implementation of an architecture that permits a **strategic and automatic reuse** is the only way they have to deliver on time and on budget. This can be done by having a methodology for managing their product variability and using a **tool suite** that supports the design, implementation, exploitation and management of Product Lines (PL). This is what the generic and open source PLUM tool is all about.

**The solution**

**Legend. Not to get lost!**
- **Decision Model**
  - Capture the variation points of a domain.
  - Define dependencies among variation points, requirements and so on.
- **Application Model**
  - An instance of the Decision Model but with the variability resolved.
  - It represents a concrete product of the family.
- **Workflow**
  - Establish product generation and deployment processes.
- **Flexible component**
  - It interprets the decision model product variability for code generation.

**The organizational issues**

Implementing a PL demands a strong organizational commitment, plus **flexible and interoperable methods and tools**, just like PLUM. A typical set up would imply:

- **A family definition process**, carried out by domain experts and engineers that have end-to-end product knowledge.
- From a **Decision Model**, a **variability resolution process** would be started by stakeholders with concrete product requirements.
- The execution engineer would be responsible for establishing the way the product is packaged (compiled, tested, deployed etc.).
- **Quality engineers** would benefit from PL metrics for decision taking.

**PL and Agile?**

The most known PL technique includes automatic code and document generation and in this way reactive PL techniques (implementing only those product line variations needed in current products) are a very promising approach to deal with changing markets. But, there are also many other facets for supporting agile in the large industry. For instance:

- **Sharing a common vision of the product family**
  - Domain knowledge encapsulation and knowledge transition enablers.
  - Decision taking based on defined dependencies.
- **Sharing technology in distributed environments**
  - All the teams talks about the same structural concepts.
  - Defined roles, responsibilities and way of working.
- **Models are active documentation in a model-centric approach**
  - It free the developers for attending to low level details.
  - Diagrams ease stakeholders communication.
- **Integrate testing in the PL architecture**
  - Automatic testing for new products could be integrated.

---

**Josetxo Vicedo and Jabier Martínez**
European Software Institute
Parque Tecnológico 204 E-48170 Zamudio, Bizkaia, Spain
jabier.martinez | josetxo.vicedo}@esi.es

The development of the PLUM Suite is partially funded by Spanish Ministry of Industry, Tourism and Trade in the framework of the FLEXI (ITEA2 06022) project.