1. Software Product Line Engineering
2. Computers are good to replicate.

Humans good to create. Put in the effort once and let machines take care of it.

1. First step is to create reusable code. SPLE goes a step further.

Emphases on recycling. Only create if necessary

1. 2 definitions – mean the same thing. Create similar systems from the same components. Going back to OOP, a component would be a class.
2. Feature model: \* OOP analogy – classes are a bit of code that implements a singular feature (units of program construction are features

\* Visually represented by means of feature diagrams

Feature interaction - behavior of the interacting features such as how their response time may be changed given the integration.

* change in the execution *behavior* of the interacting features

Variability - Problem space \* Specified by a feature model

\* Enumerative in nature, as it includes all valid variants

* Solution space \* Product variants are created according to this variability.

\* Parametric variability, i.e. Variability parameterized on features occurring in a single product variant, and configure code (from a code base) for one product variant at a time.

1. When creating software, we are presented with requirements. These transform in constraints. Representation of constraints.

Feature interaction. Planet + Earth = World

1. Comparison to manufacturing companies.

Start with product family (the family is the problem)

Code reuse is the motive