Code Re-use

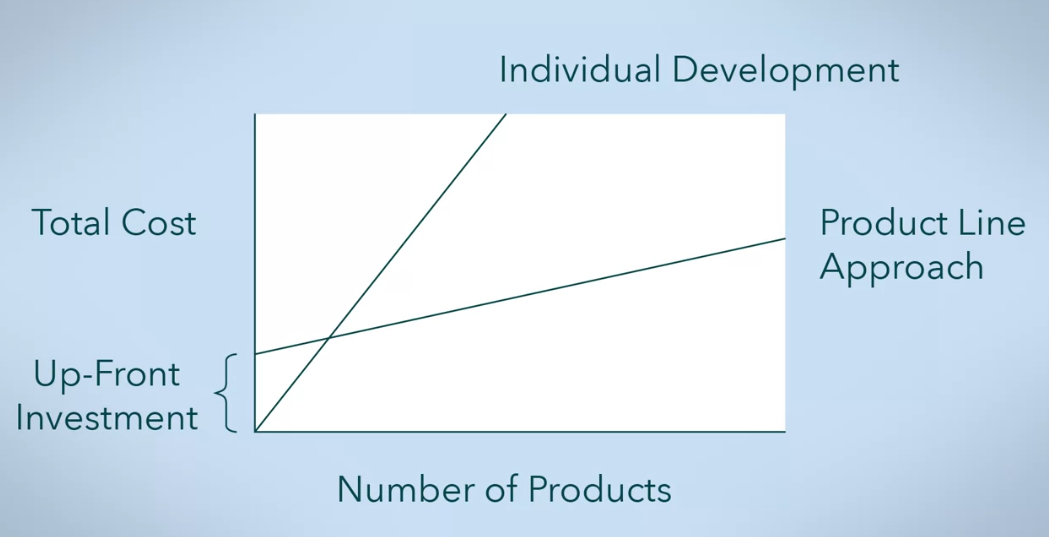
* Implementations in the real world:
  + Libraries
  + Toolkits
  + Engines
* Even if code has to be modified or adapted to the application, it is still faster than writing from scratch
* One of the many reasons that well-written and well-documented code is important

Product Line (or Product Family)

* E.g., IOS
  + iPad
  + iPod
  + iPhone

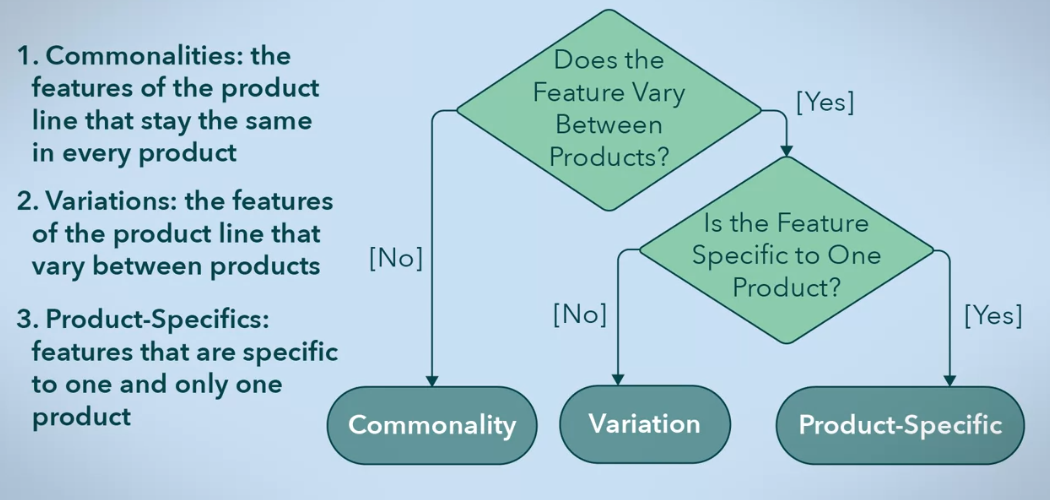
Why companies produce a Product Line?

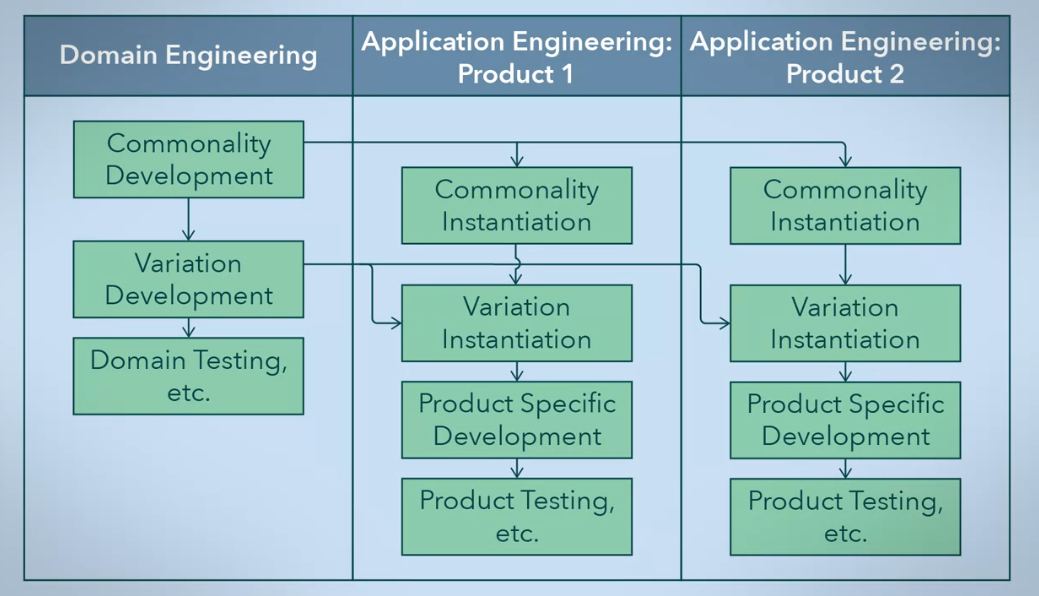
* Less cost
* More time can be spent testing for other quality attributes of the product such as security and reliability
* User Experience – less of learning curve
* Time-to-market



**Software Product Lines in Action** (By: van der Linden, Schmid, and Rommes)

* Separate the features that stay the same from the features that are different across products
  + Commonalities
    - Stay the same in every product
  + Variations
    - Vary between products
  + Product-Specifics
    - Specific to one and only one product



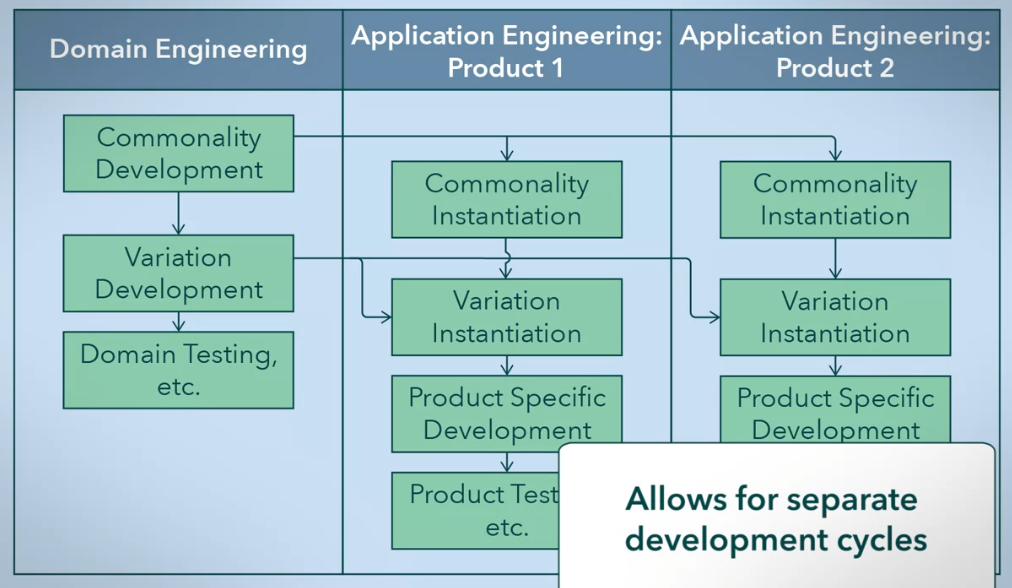


Domain Engineering

* Development of the commonalities and variations
* Putting together the building blocks of the products or “infrastructure”

Application Engineering

* Actually developing a product
* Could be several application engineering teams, one for each product
  + Using the commonalities
  + Deciding which variations are necessary
  + Integrating them into the product
  + Developing product-specific features
  + Test the final product

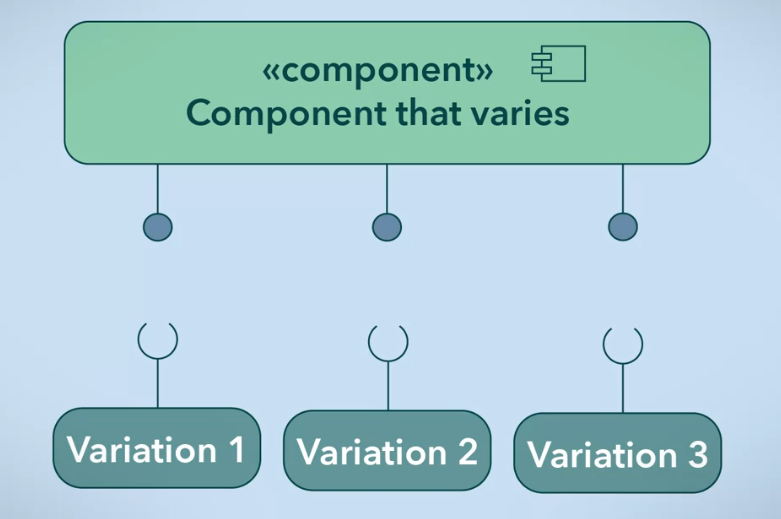


**Reference Architecture**

* The product line usually has an architecture that the products build on or change
* The domain engineering is the one responsible for this
* Must also include capacity for variation

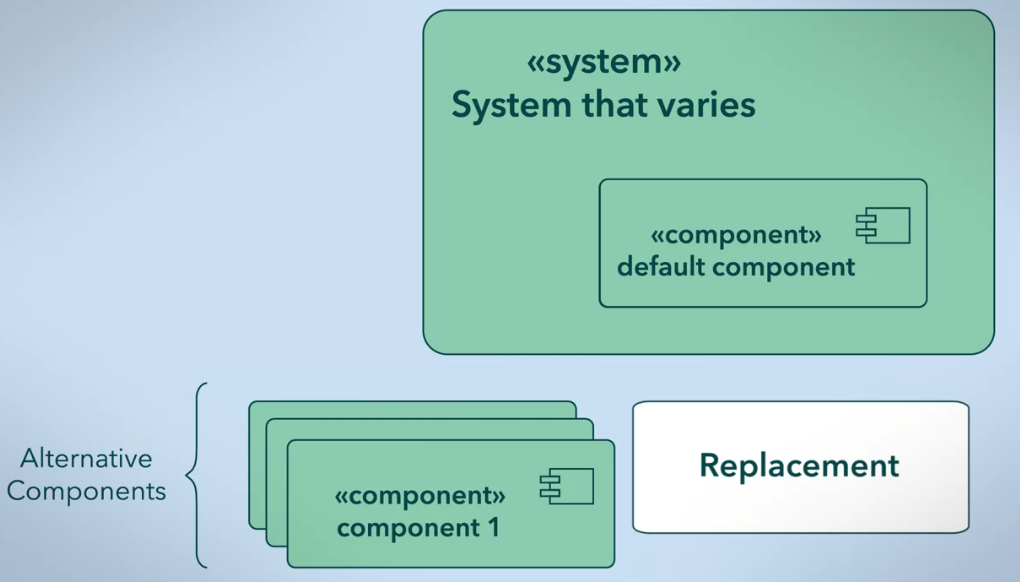
3 Techniques for Variation

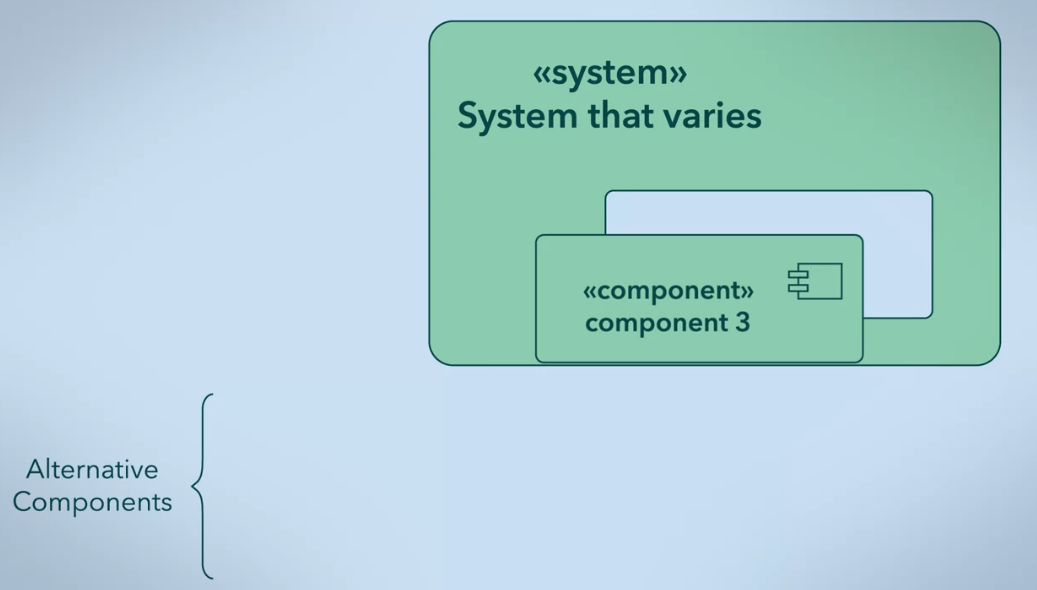
1. Adaptation



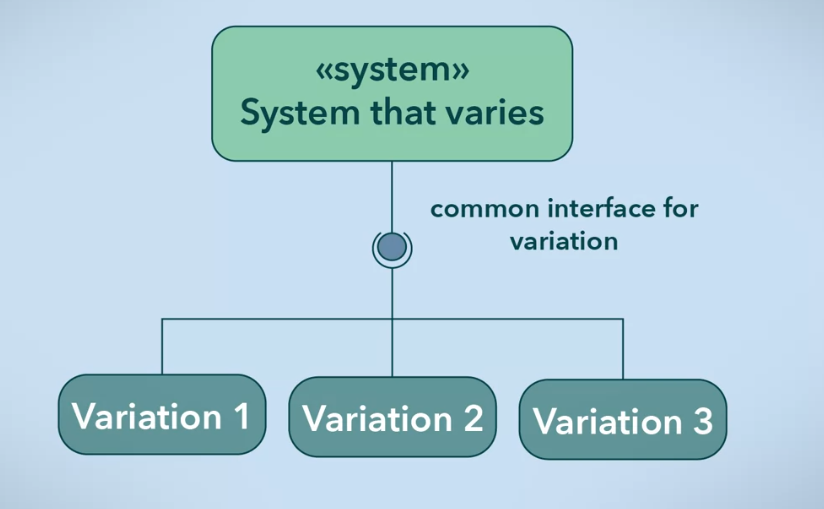
* + Only 1 implementation
  + But supplies different interfaces
    - Settings
    - Configs
    - Methods
    - Overrides

1. Replacement





1. Extension



* + Common interface is provided
    - Extensions
    - Add-ons
    - Plugins

Variations

* Can be realized in different times
  + During design or development
  + When the software is compiled or built
  + When the software is launched

Summary

* Product lines allows code reuse
* Splitting into
  + Domain Engineering
  + Application Engineering
* Product lines impose new requirements in the architecture
  + Must handle commonality and change
* Product line commonalities advantages:
  + Reliability
  + User experience
  + Security
  + Time to market
  + Maintainability
* More than 2 products, use Product Lines!