



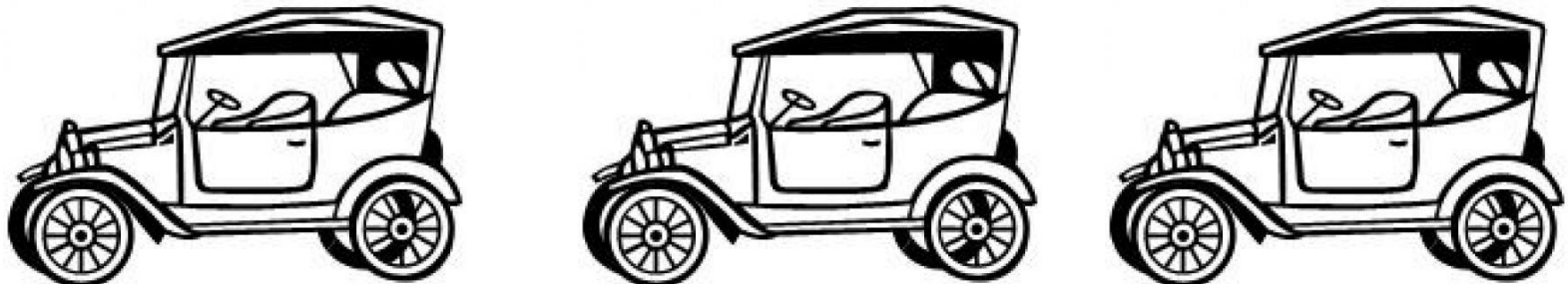
Software Product Lines

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

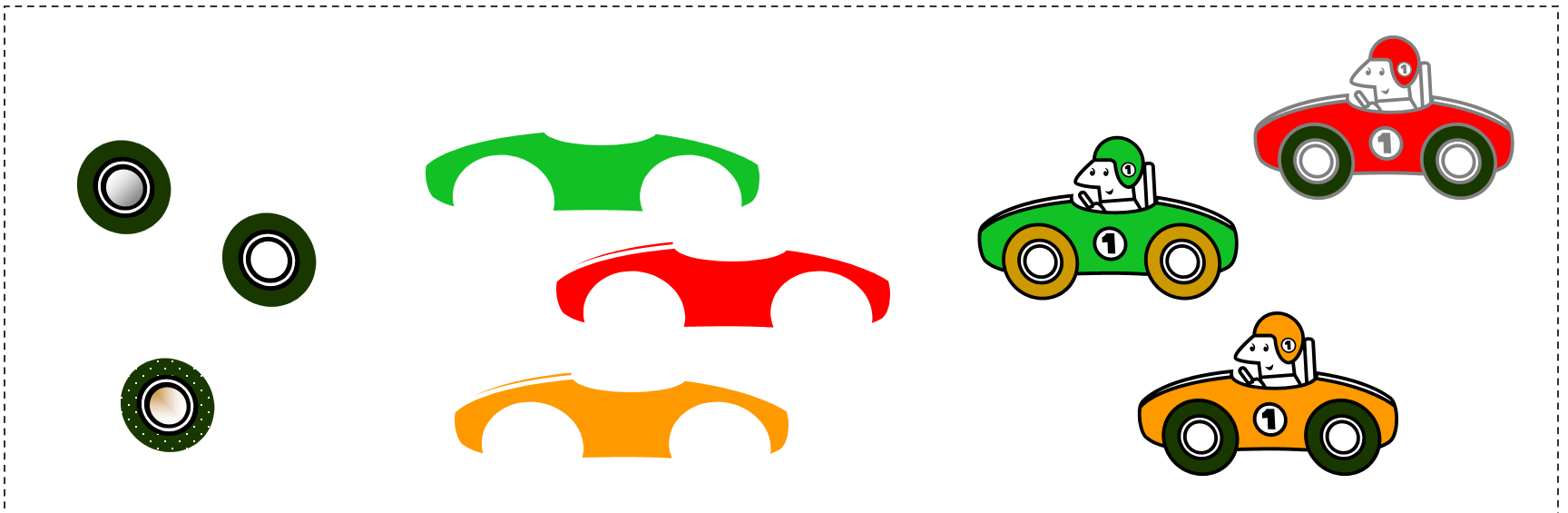
Mass Market

- Production Line (by Ford)
 - It enables production for a mass market
 - Products are cheaper than individual products
- However, production line reduces possibilities for diversification



Mass Customization

- Mass customization is the large-scale production of goods
 - It addresses individual customer's needs



[Platform and its Impact]

- A platform is any base of technologies on which other technologies are built
- In a period of three years, car companies using platforms increase sales by 35%
 - Similar pattern was observed in the camera industry (Fuji vs. Kodak)
- Product Line is *mass customization* applied to *a common platform*

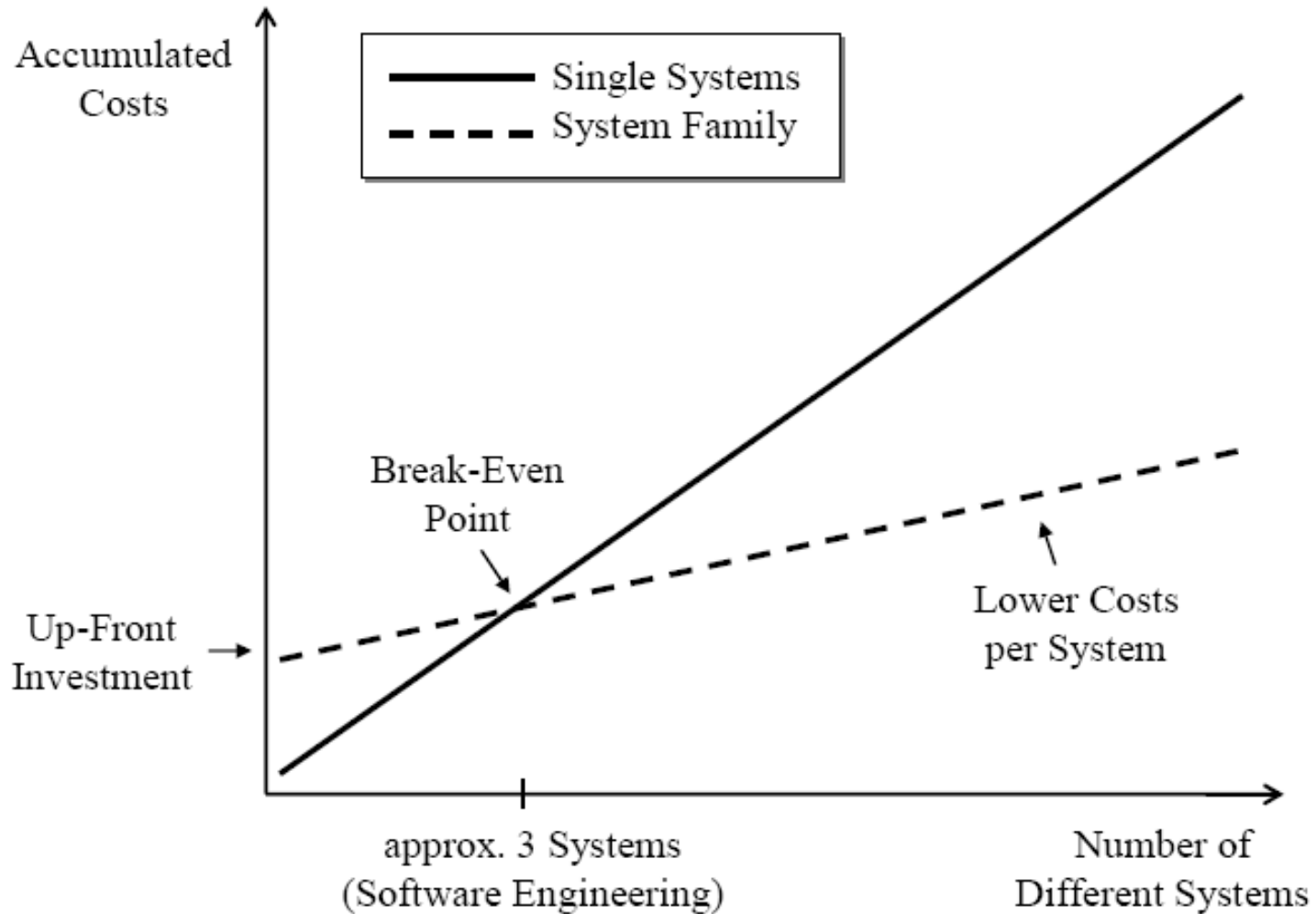
[Key Definitions]

- “A software product line (SPL) is a set of software systems that share a common set of features satisfying the specific needs of a particular market segment...” [SEI]
- Software product line engineering (SPLE) is a paradigm to develop software applications using platforms and mass customization

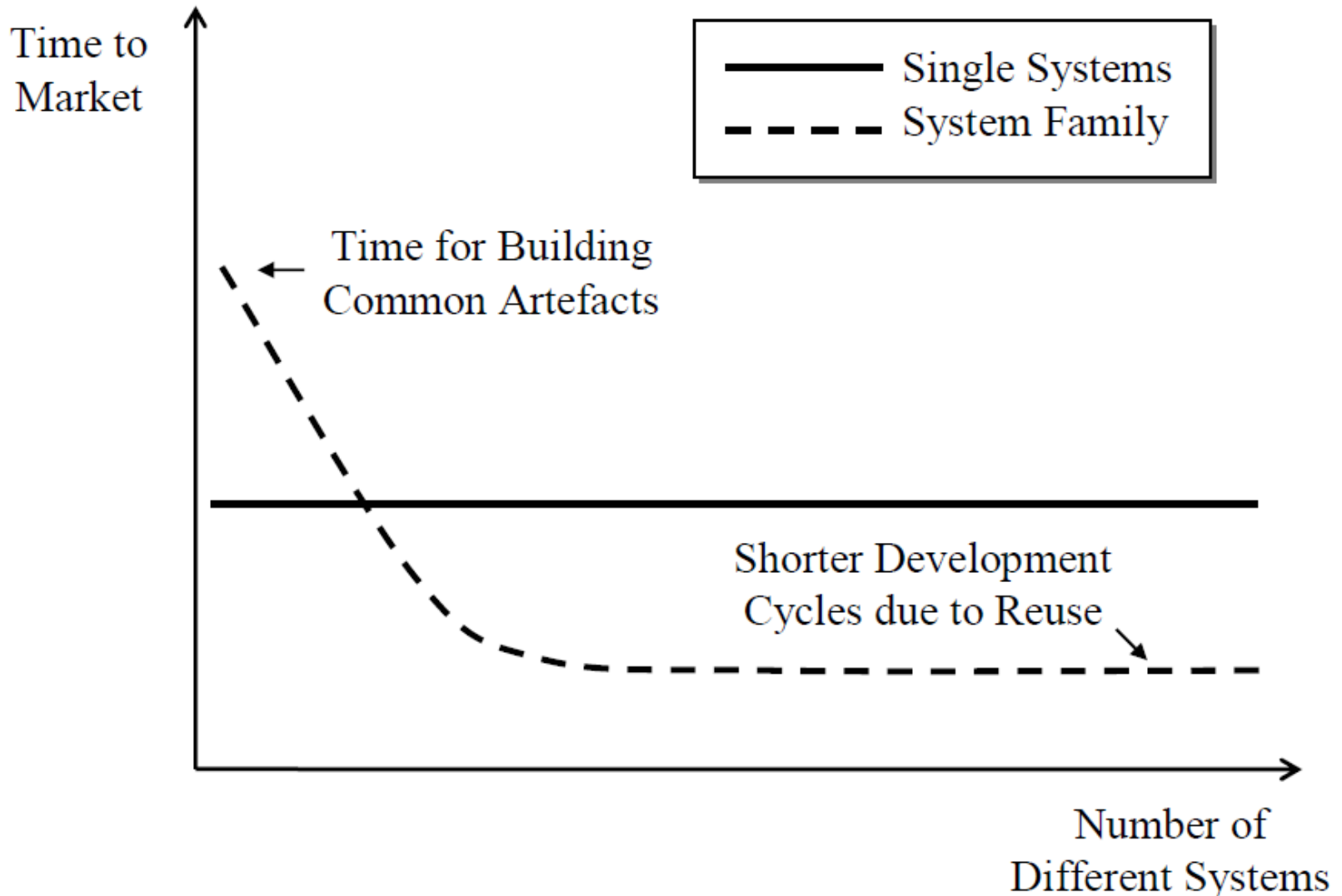
[Motivation for Adopting SPL]

- Reduction of Development Costs
- Reduction of Time to Market
- Enhancement of Quality
- Reduction of Maintenance Effort

[Lower Development Cost]



Reduction of Time to Market



[Quality and Maintenance]

- Enhancement of Quality
 - Artefact in the platform are reviewed and tested in many products
 - Extensive tests imply increasing quality of all products
- Reduction of Maintenance Effort
 - Reuse of test procedures (for instance) helps to reduce maintenance efforts

[Prerequisites for SPL]

- Product line entered in software engineering domain much later
- Prerequisites to take into consideration in the adoption of SPL
 - Enabling Technologies
 - Process Maturity
 - Domain Characteristics and Expertise

[Enabling Technologies]

- Adequate technology is required for applying SPL principles in an easy way
 - Object oriented programming is a major enabling technology
 - Component technology enables developers to package software
 - Late-binding techniques allow late configuration choices (at runtime)
- Other technologies are also desirable

[Process Maturity]

- The lack of maturity can affect the success of SPL adoption
 - Process models (such as CMMI) help to identify the weak parts of a process
- Abstraction plays a major role in making complexity manageable
 - Modelling notations (such as UML) help developers to identify variability

Domain Characteristics

- Only people who know their market can identify commonalities and variability
 - Wrong choices of variability lead to additional costs
- The stability of the domain is also important
 - If everything changes, the investment costs never pay off

[Bibliography]

- K. Pohl, G. Bockle, F. Linden.
Software Product Line Engineering: Foundations, Principles and Techniques, 1st edition, Springer, 2005.
 - Chapter 1