Software Product Lines

→ Definition

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements & Northrop, 2001].

Why does an organization want to develop a product line?

To enable systematic reuse, thereby improving software quality & productivity.

What’s your favorite product line?

Within a single organization’s boundary (intra-organizational software reuse).

What is the most important member in a software product line’s asset base?

A PLA (product line architecture).

Natural Ecosystems

Most natural ecosystems are constantly changing.
- e.g., heavy rain or prolonged dry weather
- these changes are rarely permanent & the ecosystem usually recovers over time
- e.g., human activities (waste & pollution; clearing lowland coastal forests for agriculture)

Ecosystems usually form a number of food chains (prey-predator relationships between species within an ecosystem)
Every known food chain has a base made of organisms that are able to manufacture their own food, e.g., plants.
Personas Ecosystems

Fictitious characters created to represent the different user types within a targeted demographic that might use a site or product, e.g., requirements gathering workshop.

Holistic thinking, e.g.,
What are Angie’s motivations, goals, mindsets, constraints, contexts, etc.?

Social Web Ecosystems
flickr, You Tube, typepad.

Personas

Social Web Ecosystems
flickr, You Tube, typepad.

The Four Worlds: Information Systems

A lot more worlds:
- Lawmakers (privacy, copyright)
- Managers (in time & within budget)
- Administrators
- Content/service providers
- Hackers
- Reusers

Usage System

Needs information about

Information System

Maintains information about

Development System

Builds

Contracts

The central idea of layers of technology is to build new capabilities not from scratch but rather by the addition of a new top layer that extends or specializes the capabilities of the layers below without modifying them.

Goal: Allow a diversity of technologies to co-exist and co-evolve.

Layering (Supplier-Consumer)

Application Software

Infrastructure Software (e.g., OS)

Equipment (e.g., computer, microwave, or router)

Software Components (ready-to-use elements of functionality)

Semiconductor & photonics devices (e.g., lasers & transistors)

The central idea of layers of technology is to build new capabilities not from scratch but rather by the addition of a new top layer that extends or specializes the capabilities of the layers below without modifying them.

Goal: Allow a diversity of technologies to co-exist and co-evolve.

Software Ecosystems?

→ Doesn’t make sense to some people

The term “ecosystem” implicitly suggests an attitude of non-judgmental observation (the absence of ethical judgment)

- Don’t ask how what should happen; just study and explain what does happen
- Human community is formed judgmentally (e.g., free software community)

→ Please don’t think that by putting ecosystem behind everything means you are acting in a sustainable way

How much do software engineers know ecological science? How much have they thought about sustainability?

The winning book’s view

Software industry is very complex, with many complementary products necessary to form a systems solution and complex alliances and standardization processes needed to meet the needs of numerous stakeholders. Together, the software suppliers, standardization bodies, content suppliers, service providers, and end-user organizations form a complex web of relationships. The “ecosystem” metaphor is truly descriptive.

Source: Adapted from Loucopoulos & Karakostas, 1995, p 73.
Application & Infrastructure

→ Telecommunications
  % The infrastructure chased the applications
  ➢ Define a compelling application with mass market appeal and then to coordinate the investment in application and infrastructure, while making the infrastructure fairly specialized to support that application
  ➢ E.g., Telephony (voice, fax, modem)

→ Computing
  % The applications chased the infrastructure
  ➢ Deploying a generic infrastructure that supports a diversity of applications
  ➢ E.g., Windows, Java VM

→ Regardless what chased what
  % Investments in new infrastructure (e.g., reuse libraries, AOP, BPR) have to proceed on faith that there will be successful applications to exploit new infrastructure
  ➢ In telecommunications, directly coordinated investments
  ➢ In computing, an initial suite of applications was viewed as the cost of establishing a new infrastructure

Inter-organizational Software Reuse

→ Benefits
  % Capability via integration (the integrated whole > sum of parts)
  ➢ Emergence (e.g., plane vs. wings, engines, wheels)
  ➢ Innovation accelerated via open platform
  ➢ E.g., Apple’s iPhone has about 35,000 applications as of June 2009
  % TCO (total cost of ownership) decreased
  ➢ Apple doesn’t maintain 3rd party iPhone applications
  % …

→ Challenges
  % How open can it be? (losing competitive edge?)
  % How can we coordinate?
  % Who do we trust?
  % Who can protect us?
  % …

Question: We all know software is different, e.g., from hardware or other material products. In your opinion, what’s the most distinguishing characteristic of software (compared to hardware)?

A Software Ecosystem Taxonomy

End-user programmability

End-user programmability

Application

Infrastructure (platform)

MS Excel, Mathematica
Google’s mashup editor
None so far

MS Office
Google AppEngine, Yahoo developer
Android, Symbian/Nokia Series 60

Windows, Linux, Apple OS X
Ebay, Amazon
None so far

Desktop
Internet
Mobile

Evolution of the dominant computing environment

An IT Taxonomy

Time of appearance

Technology leveraged

Processing
Storage
Communication

Algorithmic (e.g., sorting)
Document (e.g., Notepad)

Database (e.g., Access)

Publication (e.g., BBS)

Socio-technical (e.g., London Ambulance System)

Collaboration (e.g., IM)
Summary

→ "Software ecosystems"
  Terms (concepts) are socially constructed
  ∅ Peer-reviewed
  ∅ Such an analogy/metaphor has rich value in it (research questions)
  ∅ networks vs. food webs
  ∅ migration vs. invasion
  ∅ change vs. disturbance
  ∅ heterogeneity vs. biodiversity
  ∅ components vs. demography
  ∅ …

→ Is it the future?
  ∅ Can a software organization stay closed forever?
  ∅ Software research is interdisciplinary
  ∅ People (psychology, anthropology, etc.)
  ∅ Community (sociology, economy, etc.)
  ∅ Environment (ecology, climate change, etc.)
  ∅ …