Course: Software Technologies and Enterprise Architecture

WEEK 4: Software Product Lines

Lemlem Kassa(Dr.)
Addis Ababa Science and Technology University (AASTU)

April , 2026

Week-4: Software Product Lines

Contents

1. Overview of Software Product
2. Basic Software Product Line Concepts
3. Software Asset Management (SAM)

Week-4: Software Product Lines

AASTU

By the end of this unit, the student will be able to

- Understand the concept of Software Product Lines (SPL) and their purpose
- Identify the architecture and components of SPL
- Explain variability, feature models, and decision models in SPL
- Differentiate between generic and customized software products
- Compare SPL with traditional software development approaches
- Understand the concept of software reuse and its benefits
- Describe binding times and variation points in SPL development
- Explain the role of Software Asset Management (SAM) in managing software resources
- Identify benefits, challenges, and real-world applications of SPL

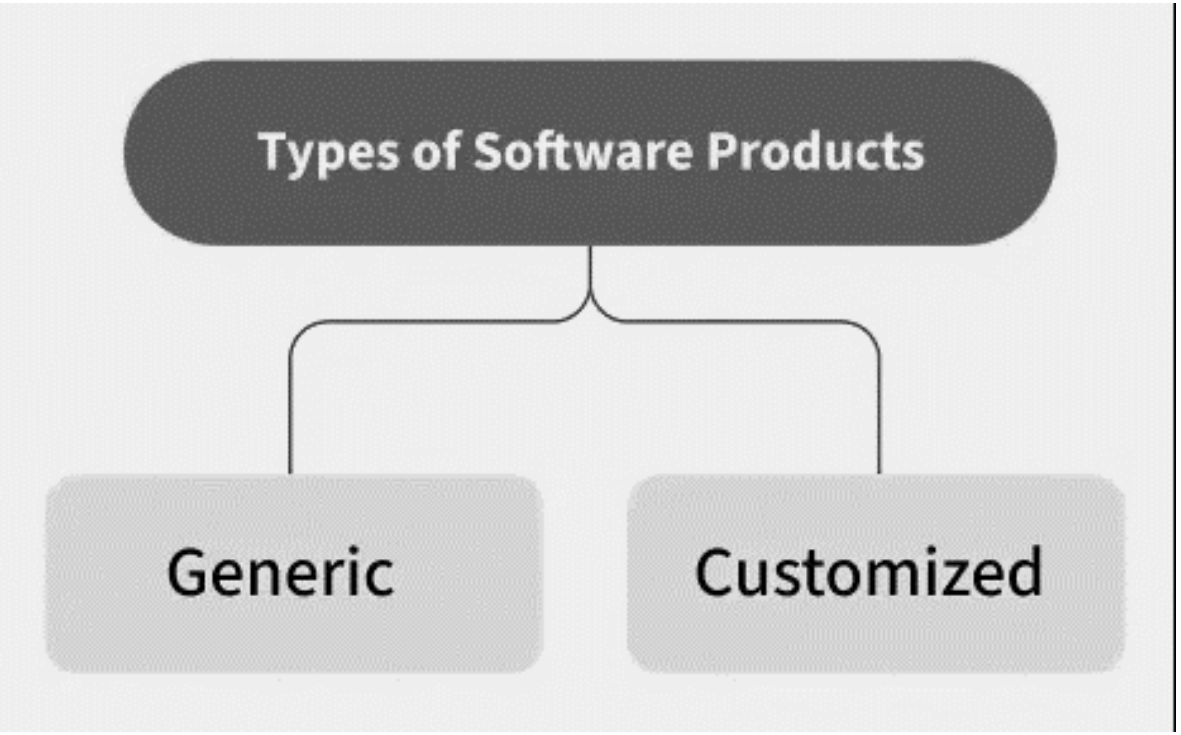
1. Overview of Software Product

- **Software Products** are any Software or any system that has been developed, tested, and maintained for the specific purpose to solve that problem.
- In certain cases, software products may be part of system products where hardware, as well as software, is delivered to a customer.
- Produced with the help of the software process.
- The software process is a way in which we produce software.

[1]. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/

Cont.’. …1. Overview of Software Product

- Based on why the software product are developed, the Software product mainly divided into the two parts which are follows.



[1]. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/

Cont.'. …1. Overview of Software Product

Generic products: stand-alone systems that are developed by a production unit and sold on the open market to any customer who can buy them.

- They are ready to use right out of the box and are designed to meet the general needs of many users.
- Whether we're working on a document or browsing the web, these tools are flexible enough to suit a wide range of tasks.

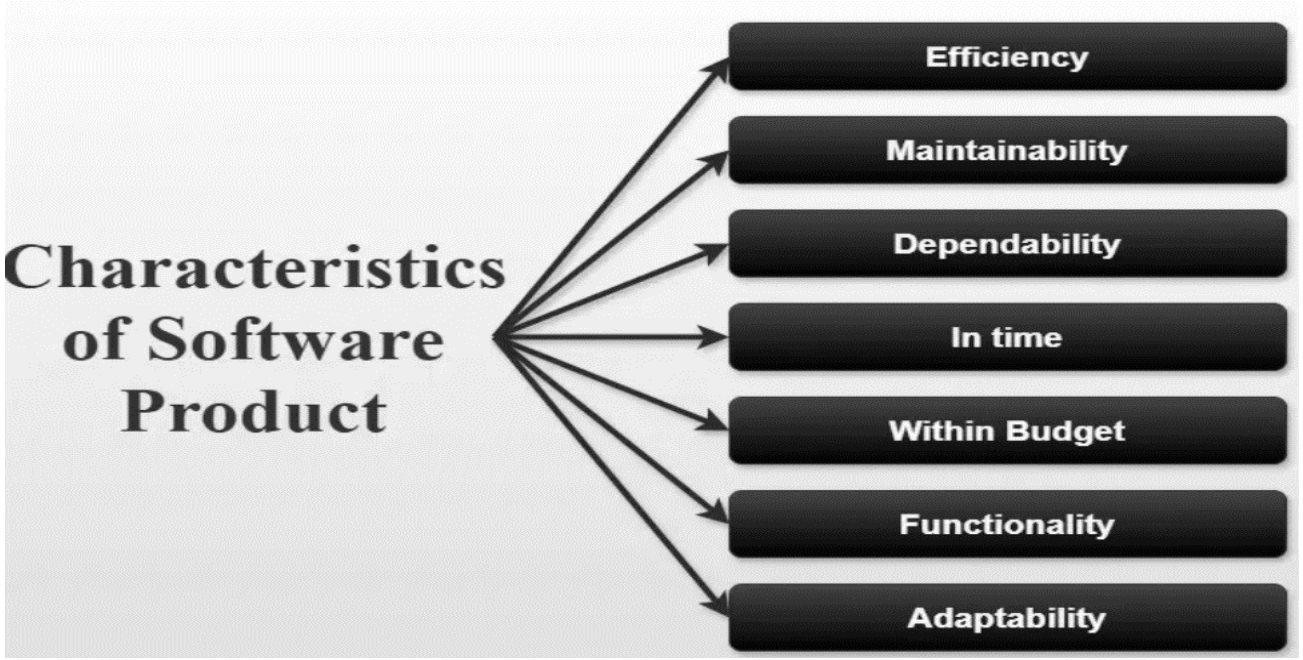
Example:

- Products like Microsoft Office or Google Chrome.

Cont.’. …1. Overview of Software Product

Characteristics of Software Product

- Software Product should possess the following important characteristics:



[1]. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/

Cont.'. …1. Overview of Software Product

Characteristics of Software Product

1. **Efficiency:** The software should not make wasteful use of system resources such as memory and processor cycles.
2. **Maintainability:** It should be possible to evolve the software to meet the changing requirements of customers.
3. **Dependability:** It is the flexibility of the software that ought not to cause any physical or economic injury in the event of system failure. It includes a range of characteristics such as reliability, security, and safety.

[1]. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/

Cont.'. …1. Overview of Software Product

Cont. ..Characteristics of Software Product

4. In time: Software should be developed well in time.

5. Within Budget: The software development costs should not be overrun, and they should be within the budgetary limit.

6. Functionality: The software system should exhibit the proper functionality, i.e., it should perform all the functions it is supposed to perform.

7. Adaptability: The software system should have the ability to adapt to a reasonable extent with the changing requirements.

Cont. …1. Overview of Software Product

2. Customized Products: systems that are commissioned by a particular customer.

- Some contractor develops the software for that customer.
- These are designed specifically for a particular business or set of users.
- A software system built just for managing employee data in a company or a custom CRM system that's perfectly aligned with the way a business operates.

https://share.google/gjTt9Og8Rm1ZLqPKP

[1]. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/

Cont. …1. Overview of Software Product

Software Products Vs Software Services

1. Software Products: a type of software system, i.e., outcome of the software development process, and are delivered to customers along with documentation about how to install and use the particular software system.

2. Software Services: a type of software distribution model which is used to deliver applications over the internet simply as a service, and allows data to be accessed from any device using the internet.

[2]. Difference between Software Products and Software Services, GeeksforGeeks, GeeksforGeeks, 2024,https://www.geeksforgeeks.org/software-engineering/difference-between-software-products-and-software-services/

Cont. …1. Overview of Software Product

Software Products Vs Software Services

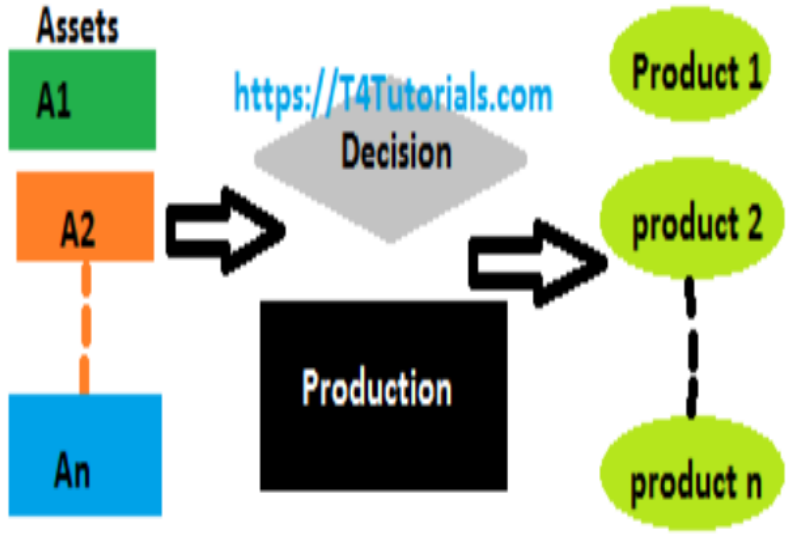
Software Products	Software Services
Software Products generally represent high end work that is done by vendors.	Software Services generally represent low end work that is done by vendor.
Products includes activities that add higher value than activities of services.	Services includes activities that add lower value than activities of products.
Software product companies usually develop products.	Software service companies may or may not develop products.
Features of software products include efficiency, maintainability, adaptability, etc.	Features of software services include application security, high availability, data security, etc.
Types of software product includes generic products, customized products, etc.	There are no such type of software services.
List of software products include Outreach, Zoho Desk, Freshdesk, Intercom, etc.	List of software services includes IT Security services, infrastructure support services, app maintenance services, etc.
Benefits of software product companies includes provide right product to market, control schedule and cost, focus on customer requirements, etc.	Benefits of software service companies includes flexibility, reduce time to benefit, easy to use, requires lower cost, etc.
Number of labors required is not much high but should be talented and expert.	Number of labors required is very high for increasing effort and its not necessary that they should be highly skilled.

[2]. Difference between Software Products and Software Services, GeeksforGeeks, GeeksforGeeks, 2024,https://www.geeksforgeeks.org/software-engineering/difference-between-software-products-and-software-services/

Cont. …1. Overview of Software Product

What is a software product line?

- A collection of multiple software.
- Software engineers can share different assets among all the software available in the respective product line.
- We can share the common, managed set of software features that can satisfy the specific needs of a big task.
- These common features are developed from a common set of core assets.



Software product line SPL

[3]. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/

2. Basic Software Product Line Concepts

- **Software asset inputs**: a collection of software assets such as *requirements, source code components, test cases, architecture, and documentation* that can be configured and composed in different ways to create all of the products in a product line.
- Each of the assets has a well-defined role within a common architecture for the product line.
- To accommodate variation among the products, some of the assets may be optional, and some of the assets may have internal *variation points* that can be configured in different ways to provide different behavior.

[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools. https://www.methodsandtools.com/archive/archive.php?id=45

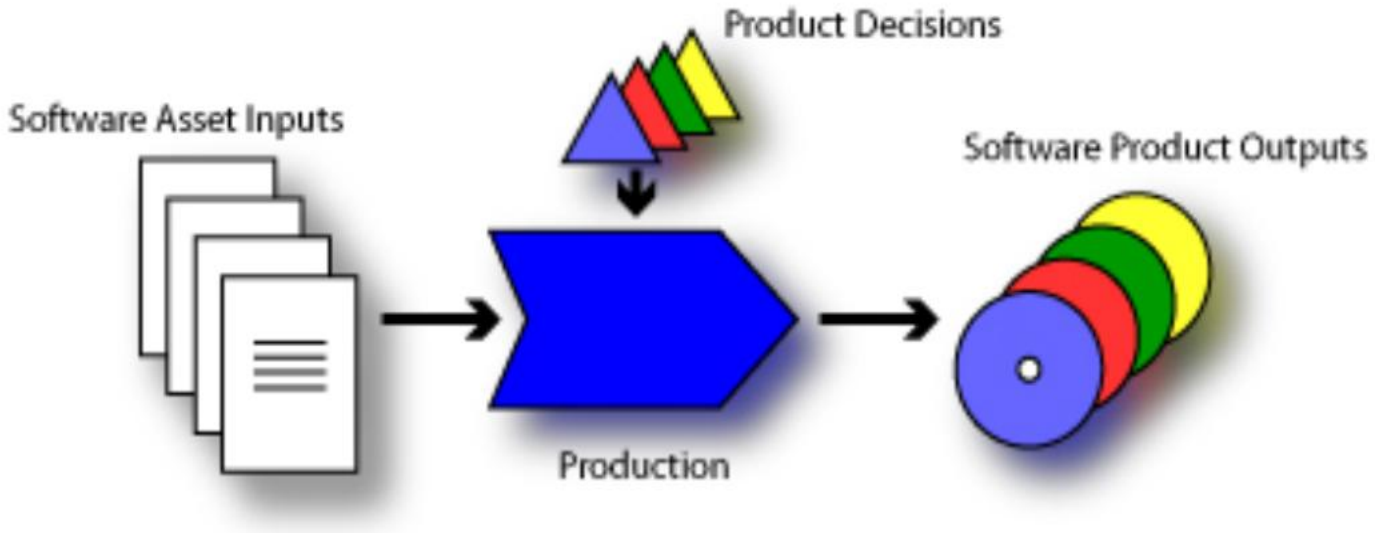
Cont.' ..2. Basic Software Product Line Concepts

- **Decision model and product decisions**: The *decision model* describes optional and variable features for the products in the product line.
  - Each product in the product line is uniquely defined by its *product decisions* - choices for each of the optional and variable features in the decision model.
- **Production mechanism and process**: the means for composing and configuring products from the software asset inputs.
  - Product decisions are used during production to determine which software asset inputs to use and how to configure the variation points within those assets.

[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools. https://www.methodsandtools.com/archive/archive.php?id=45

Cont.' ..2. Basic Software Product Line Concepts

- **Software product outputs:** the collection of all products that can be produced for the product line.
- The *scope* of the product line is determined by the set of software product outputs that can be produced from the software assets and decision model.



[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools. https://www.methodsandtools.com/archive/archive.php?id=45

Cont.’ ..2. Basic Software Product Line Concepts

Key objectives of software product lines

- To capitalize on commonality and manage variation in order to reduce the time, effort, cost, and complexity of creating and maintaining a product line of similar software systems.
  - *Capitalize on commonality*:- through consolidation and sharing within the software asset inputs, thereby avoiding duplication and divergence.
  - *Manage variation*:- by clearly defining the variation points and decision model, thereby making the location, rationale, and dependencies for variation explicit.

[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools. https://www.methodsandtools.com/archive/archive.php?id=45

Cont.' ..2. Basic Software Product Line Concepts

Software reuse?

- The important content of the Software Product Line
- It is a term used for developing the software by using the existing software components. Some of the components that can be reused are as follows;
  - Source code
  - Design and interfaces
  - User manuals
  - Software Documentation
  - Software requirement specifications, etc.

[3]. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/

Cont.' ..2. Basic Software Product Line Concepts

Advantages of software reuse

- **Less effort:** Software reuse requires less effort because many components use in the system are ready made components.
- **Time-saving:** Re-using the ready made components is time saving for the software team.
- **Reduce cost:** Less effort, and time saving leads to the overall cost reduction.

[3]. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/

Cont.’ ..2.Basic Software Product Line Concepts

… Advantages of software reuse

- **Increase software productivity:** when you are provided with ready made components, then you can focus on the new components that are not available just like ready made components.
- **Utilize fewer resources:** Software reuse save many sources just like effort, time, money etc.
- **Leads to a better quality software:** Software reuse save our time and we can consume our more time on maintaining software quality and assurance.

[3]. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/

Cont.’ ..2. Basic Software Product Line Concepts

Software reuse success factors

1. Capturing Domain Variations
2. Easing Integration
3. Understanding Design Context
4. Effective Teamwork
5. Managing Domain Complexity

[3]. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/

Cont.' ..2.Basic Software Product Line Concepts

Binding Times

- The primary distinction between **software product line engineering** and **conventional software engineering** is the presence of variation in some or all of the software assets.
- In the early stages of a software product line lifecycle, software assets contain *variation points* that represent unbound options about how the software will behave.
- At some point during the production process, product decisions are utilized to select among the options for each variation point, after which the behavior of the variation point in the final product is fully specified.
- The time at which the decisions for a variation point are bound is referred to as the ***binding time***.

[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools.
https://www.methodsandtools.com/archive/archive.php?id=45

Cont.’ ..2. Basic Software Product Line Concepts

Examples of **different binding times for software product lines** include:

- *Source reuse time.* Decisions bound when reusing a configurable source artifact
- *Development time.* Decisions bound during architecture, design, and coding
- *Static code instantiation time*. Decisions are bound during the assembly of code just before a build
- *Build time*. Decisions bound during compilation or related processing
- *Runtime*. Decisions bound when the system is executing

[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools.
https://www.methodsandtools.com/archive/archive.php?id=45

Cont. ..2. Basic Software Product Line Concepts

…..**Examples** of **different binding times for software product lines include**:

- *Package time*. Decisions bound while assembling binary & executable collections
- *Customer customizations*. Decisions bound during custom coding at customer site
- *Install time*. Decisions bound during the installation of the software product
- *Startup time*. Decisions bound during system startup

Cont. ..2. Basic Software Product Line Concepts

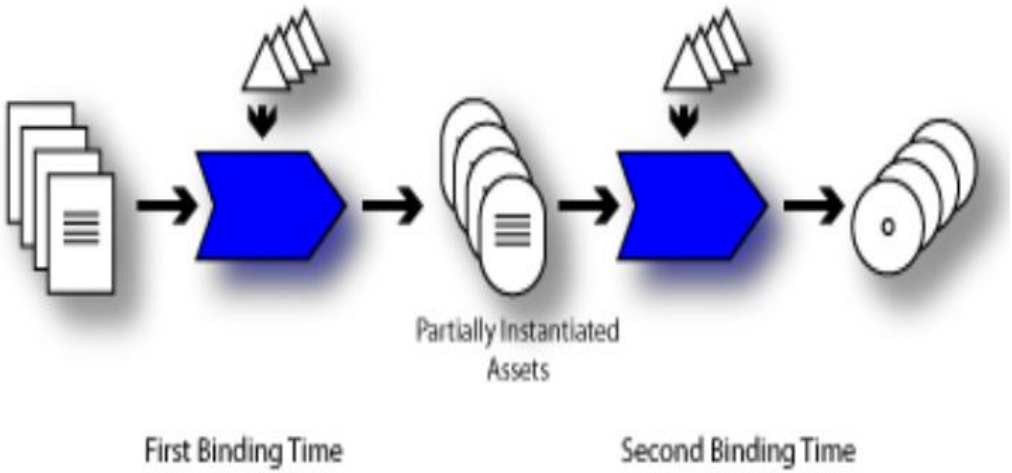
- It is possible to utilize multiple binding times in a software product line.
- This allows some decisions to be bound earlier in the lifecycle and other decisions to be deferred until later in the process.

Example,

- Some decisions might best be made by a product manager at the company developing the software, while other decisions might best be made by the end customer who will use the software.

Cont. ... 2. Basic Software Product Line Concepts

- With multiple binding times, the software product outputs from binding decisions at one production stage become *partially instantiated software* asset inputs for binding decisions at the next production stage.
- The figure below illustrates two binding times, though more are possible.



[4]. Estimating With Use Case Points, Mike Cohn, Methods & Tools.
https://www.methodsandtools.com/archive/archive.php?id=45

3. Software Asset Management (SAM)

- Software asset management (SAM) is a set of best practices and processes that help organizations efficiently manage, control, and protect their software assets throughout their lifecycle.
- This includes tracking and managing software licenses, ensuring compliance with licensing agreements, optimizing software usage, and reducing costs associated with software procurement and maintenance.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

3. Software Asset Management (SAM)

Software Asset Management Tools

- The increase in software usage across various departments and the shift towards cloud computing have made SAM more vital than ever.
- Software asset management tools are software solutions designed to automate the management and optimization of software assets and licenses within an organization.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Cont.. …3. Software Asset Management (SAM)

Software Asset Management Tools

- **Inventory Management:** Automatically collect data on all installed software applications across your network, ensuring a comprehensive and up-to-date inventory.
- **License Management**: SAM tools track the number of licenses purchased versus those in use to ensure compliance with licensing agreements and prevent both under-licensing and over-licensing.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Cont.. …3. Software Asset Management (SAM)

…Software Asset Management Tools

- **Compliance Assurance:** Monitor software usage and license terms to help your organization steer clear of legal repercussions associated with non-compliance.
- **Cost Management and Usage Analytics**: Identify unused or underutilized licenses to make informed decisions about software needs, potentially reallocating or terminating unnecessary licenses to reduce costs.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Cont.. …3. Software Asset Management (SAM)

….Software Asset Management Tools

- **Contract Management:** Manage contracts and documentation associated with software purchases and licenses, ensuring all obligations are met and accessible.
- **Enhanced Security:** SAM tools help identify and mitigate security risks associated with outdated or unsupported software.

By integrating SAM tools, organizations can enhance operational efficiency, reduce compliance risks, and manage software costs more effectively.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Cont.. ...3. Software Asset Management (SAM)

Best Practices for Software Asset Management

- Successfully implementing a SAM program involves adhering to established best practices.

1. Establish Clear Policies and Procedures

- Define clear policies and procedures for software procurement, deployment, and usage. Ensure that all employees are aware of these policies and understand their roles in maintaining compliance.

2. Conduct Regular Audits

- Regularly audit your software assets to ensure compliance with licensing agreements and identify any discrepancies.
- Use automated SAM tools to streamline the audit process and maintain accurate records.

Cont.. …3. Software Asset Management (SAM)

3. Optimize Software Licenses

- Analyze software usage patterns to identify underutilized licenses. Reallocate or retire unused licenses to optimize costs and improve efficiency.

4. Stay Informed About Licensing Changes

- Stay updated on changes in software licensing agreements and vendor policies. Ensure that your SAM program is adaptable to accommodate these changes and maintain compliance.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Cont. …3. Software Asset Management (SAM)

5. Leverage SAM Tools

- Use advanced SAM tools to automate and streamline the management of our software assets. These tools provide valuable insights and help maintain compliance with minimal manual effort.
- To get the most from these best practices, organizations need a platform that combines visibility, automation, and intelligent insights.
- This transforms SAM from a reactive necessity into a proactive strategy for efficiency and control.

[5]. Software Asset Management (SAM), Xensam, Xensam, 202.https://xensam.com/resources/blog/software-asset-management/

Summary

- Software products are developed to solve specific problems and are classified into generic and customized products. Generic products are ready-made for general users, while customized products are built for specific clients
- Software Product Lines (SPL) are collections of related software systems that share common features and core assets. SPL focuses on reusing software components such as code, design, and documentation to improve efficiency
- Core SPL concepts include software assets, decision models, production processes, and product outputs. Variability management allows different products to be created by configuring variation points
- Binding time defines when decisions about variation are made during the development lifecycle
- Software reuse reduces development time, cost, and effort while improving productivity and quality. Successful SPL implementation requires effective teamwork, domain understanding, and integration
- Software Asset Management (SAM) helps manage software licenses, usage, compliance, and costs. SAM tools support inventory management, license tracking, cost optimization, and security
- Best practices in SAM include regular audits, clear policies, and efficient license management

References

1. Software Product, GeeksforGeeks, GeeksforGeeks, 2023. https://www.geeksforgeeks.org/software-engineering/software-engineering-software-product/
2. Difference between Software Products and Software Services, GeeksforGeeks, GeeksforGeeks, 2024,https://www.geeksforgeeks.org/software-engineering/difference-between-software-products-and-software-services/
3. Software Product Line Examples, T4Tutorials, T4Tutorials, 2021. https://t4tutorials.com/software-product-line-examples/.
4. Estimating With Use Case Points, Mike Cohn, Methods & Tools. https://www.methodsandtools.com/archive/archive.php?id=45
5. Software Asset Management (SAM), Xensam, Xensam, Jan. 2026. https://xensam.com/resources/blog/software-asset-management/


Thank you!