



Universidad Veracruzana

EMPLEO DE *CONCERNS* PARA ARQUITECTURAS DE LÍNEAS DE PRODUCTOS DE SOFTWARE

Taller de Líneas de Productos de Software
13 al 15 de Octubre de 2010
Dra. María Karen Cortés Verdín

AGENDA



- Antecedentes
- *Concerns* (Intereses)
- AOPLA
- Conclusiones

ANTECEDENTES



- ◉ ¿Qué es una Línea de Productos de Software (LPS)?

“conjunto de sistemas de software intensivo que comparten un conjunto administrado de características comunes que satisfacen las necesidades específicas de un segmento del mercado o misión y que se desarrollan a partir de un conjunto de activos esenciales en una maneta prescrita.”¹

1. P. Clements, and L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley, USA, 2001

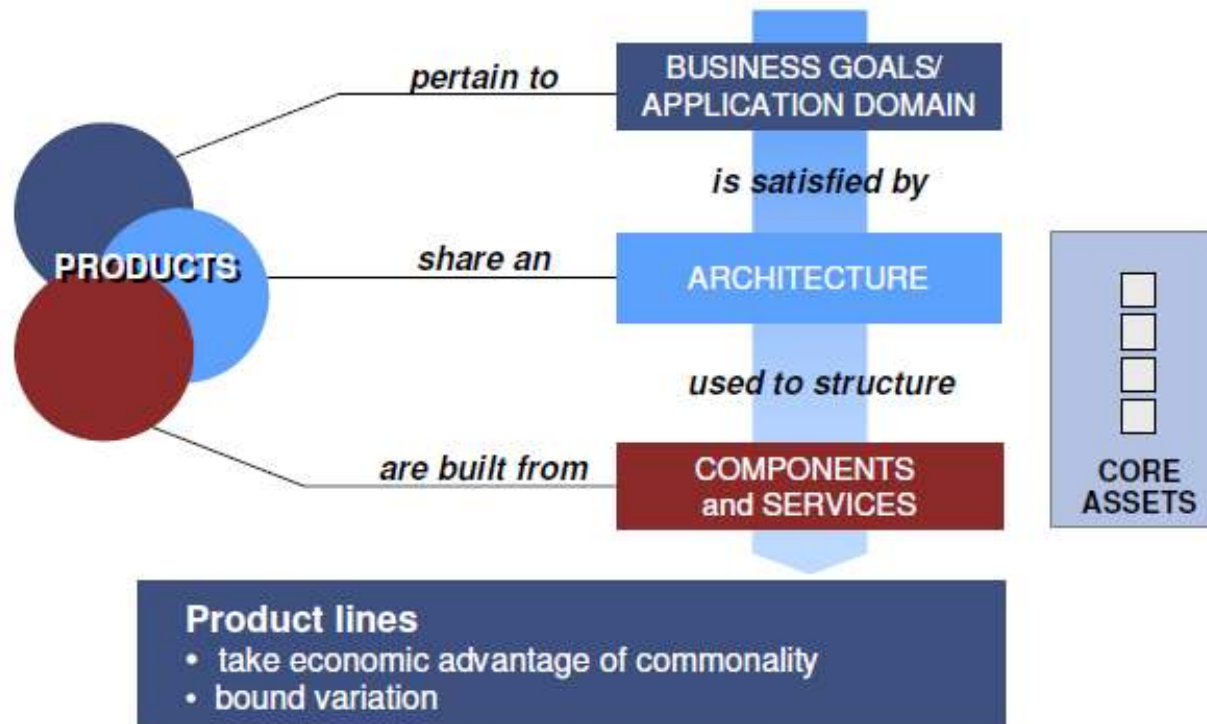


- ¿De qué se ocupa la Ingeniería de LPS?
 - Falta de satisfacción con el desempeño actual del proyecto/producto
 - Necesidad de reducir el costo y tiempo
 - Complejidad al administrar y mantener demasiadas variantes de productos
 - Falta de personal
 - Necesidad de responder rápidamente a las demandas del mercado/cliente



En realidad, estos problemas NO son nuevos . .
. ¿Entonces?







Feed control and farm management software



Bold Stroke Avionics

E-COM Technology Ltd.

Medical imaging workstations

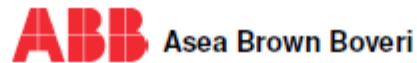


Firmware for computer peripherals



Lucent Technologies
Bell Labs Innovations

5ESS telecommunications switch



Gas turbines, train control, semantic graphics framework



Internet payment gateway infrastructure products



AXE family of telecommunications switches



Elevator control systems



Mobile phones, mobile browsers, telecom products for public, private and cellular networks



Computer printer servers, storage servers, network camera and scanner servers



Customized solutions for transportation industries



Software for engines, transmissions and controllers



RAID controller firmware for disk storage units



Interferometer product line



PHILIPS

High-end televisions,
PKI telecommunications switching
system, diagnostic imaging equipment

**Rockwell
Collins**

Commercial flight control system avionics,
Common Army Avionics System (CAAS),
U.S. Army helicopters

symbian

EPOC operating system



Test range facilities

RICOH

Office appliances

SALION

TARGET. WIN. DELIVER.

Revenue acquisition
management systems

TELVENT

Industrial supervisory control
and business process
management systems




Command and
control simulator for
Army fire support

BOSCH 

Automotive gasoline
systems

SIEMENS

Software for viewing and
quantifying radiological images

 Climate and flue gas
measurement devices

Support software

 **MOTOROLA**

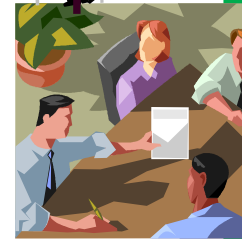
Pagers product line

Northrop Linda M. Software Product Lines Essentials



Principales características de una LPS:

- Organización del desarrollo de software en dos partes
- Reutilización planeada y proactiva de los activos esenciales
- Desarrollo centrado en la arquitectura





- La arquitectura de software es la portadora principal de la calidad del sistema (rendimiento, seguridad, facilidad de mantenimiento, etc.) Ninguno puede lograrse sin una visión arquitectónica unificada.
- En un enfoque de LPS la arquitectura es aún más importante.





Consideraciones en PLA (Arquitecturas de LP)

- Definición de una arquitectura de referencia
 - Una arquitectura esencial que captura el diseño de alto nivel de las aplicaciones de la LPS²
- Atributos de calidad de los productos así como atributos de calidad específicos de la LP
- Soporte para la evolución
- Similitudes y variación

2. Pohl Klaus, Bockle Gunter, van der Linden Frank, Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag. 2005.

INTERESES



- Los *concerns* son . . . “aquellos intereses que pertenecen al desarrollo del sistema, su operación o cualquier otro de los aspectos que son críticos o importantes para uno o más de los interesados”³

3. IEEE Std. 1471- 2000. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.



- La separación de intereses es un principio de diseño que ha sido empleado ampliamente en el desarrollo de sistemas
- A pesar de ello, los intereses no han sido considerados entidades de primera clase
- En AOPLA el modelado de intereses es un elemento fundamental



Los intereses surgen en cada etapa del ciclo de vida, abarcando actividades, artefactos, métodos y herramientas

- Facilidad de uso
- Disponibilidad
- Correctez . . .



- Funcionalidad
- Integridad de datos
- Seguridad
- Disponibilidad . . .



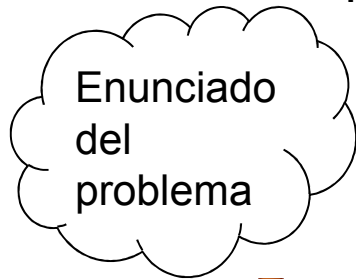
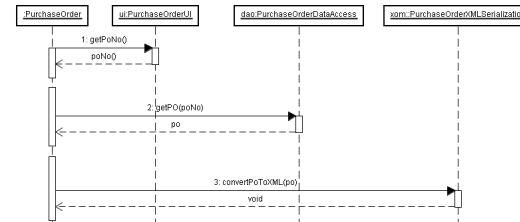
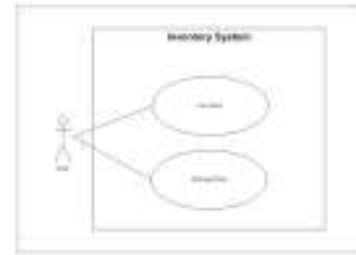
- Requerimientos
- Correctez
- Facilidad de aprender . . .



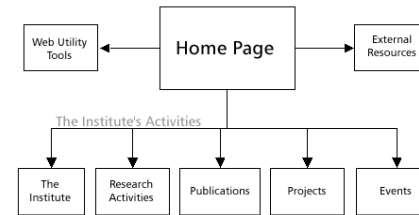
- Facilidad de evolución
- Facilidad de modificación
- Modularización . . .



Rendimiento

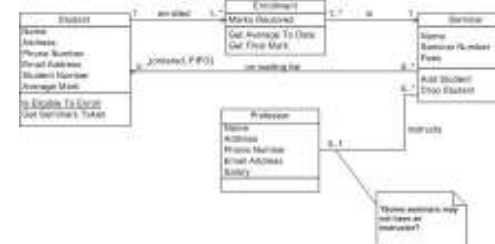
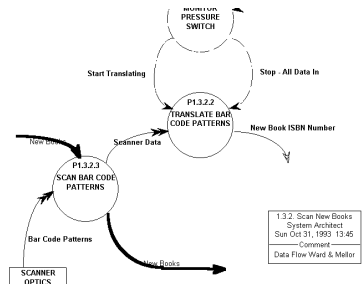


Funcionalidad 1



...

Funcionalidad n



Facilidad de mantenimiento





- *Tyranny of the dominant decomposition:*
 - “ el programa sólo puede modularizarse de una manera a la vez, y los muchos tipos de intereses que no se alínean a dicha modularización quedan dispersos en muchos módulos y entrelazados entre sí”⁴

- Separando adecuadamente los intereses, el software puede ser más fácil de evolucionar, mantener, entender, personalizar y reutilizar

4. <http://researchweb.watson.ibm.com/morphogenic/>



◉ Cosmos (Concern Space Modeling Schema)

- ◉ Es un esquema para modelado de espacios de intereses con las siguientes características:
 1. Soporta el modelado de intereses (vista multidimensional)
 2. Es independiente de lenguaje y metodología
 3. Se acomoda a los diversos formalismos empleados a lo largo del ciclo de vida
 4. Permite representar cualquier tipo de interés
 5. Permite representar cualquier tipo de relación entre intereses



○ Intereses en Cosmos

Core			Extensions
Concerns	Logical	Kinds, Instances, Properties, Topics	
	Physical	Collections, Instances, Attributes	
Relationships	Categorical	ClassOf, InstancesOf, PropertyOf, SubjectOf, MemberOf, AttributeOf	
	Interpretive	SignificantFor	Admits, ContributesTo, LogicallyImplements, Motivates, LogicallyPartOf
	Mapping	Maps to	AffectedBy, Describes, Models, PhysicallyImplementedBy, Represents
	Physical	PhysicallyRelatesTo	Connects, ConnectsTo, Physically Affects, PhysicallyPartOf
Predicates	<no subtypes defined>		



- Los predicados son el área más abierta de Cosmos
- Aunque la consistencia es algo importante, no es posible incluir tipos de predicados por la variabilidad que puede haber en los diversos espacios de intereses
- Los tipos de predicados se pueden agregar por extensión



ASPECT-ORIENTED PRODUCT LINE ARCHITECTURE (AOPLA)

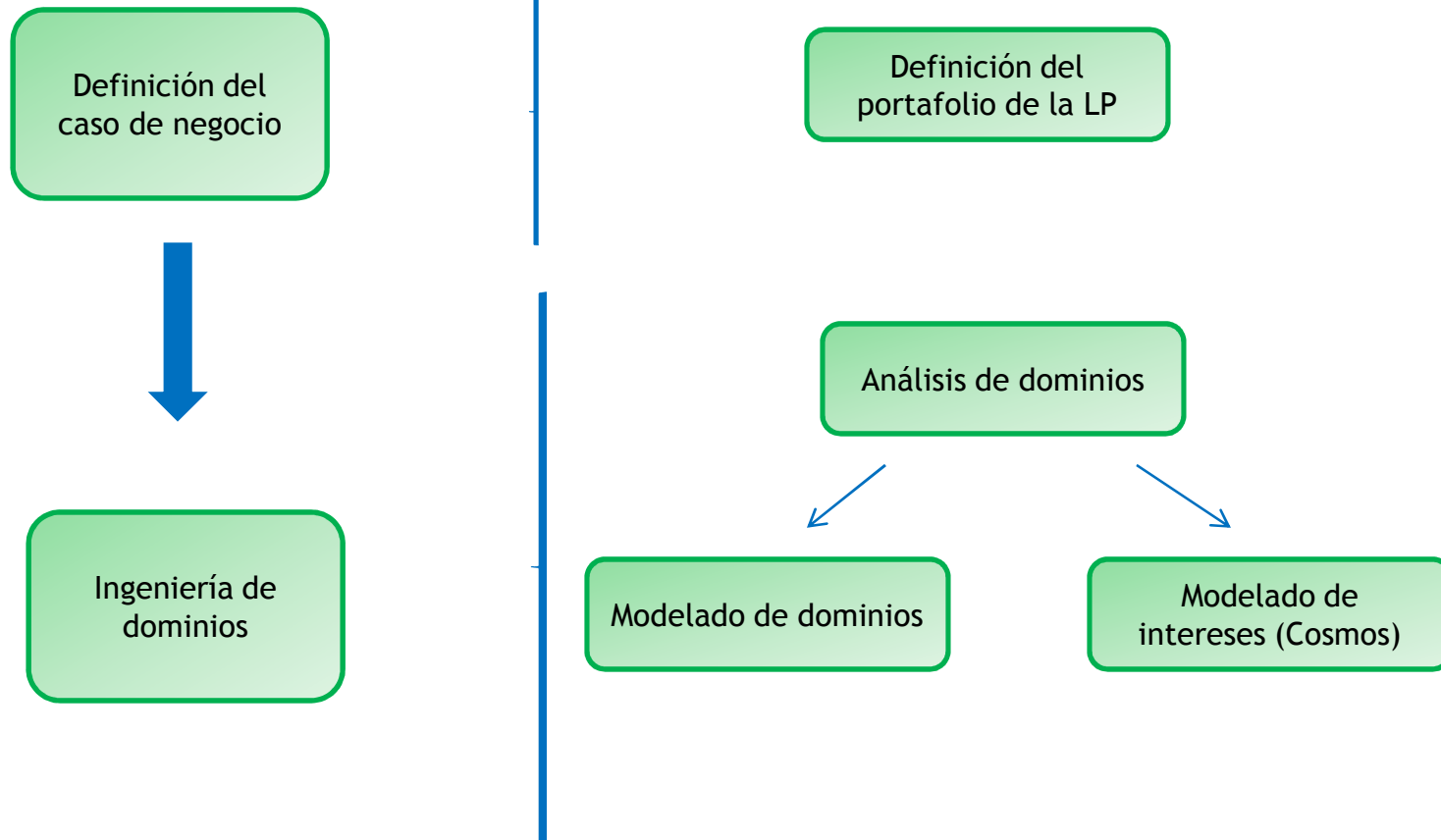
Un enfoque para el diseño de PLA que incluye:

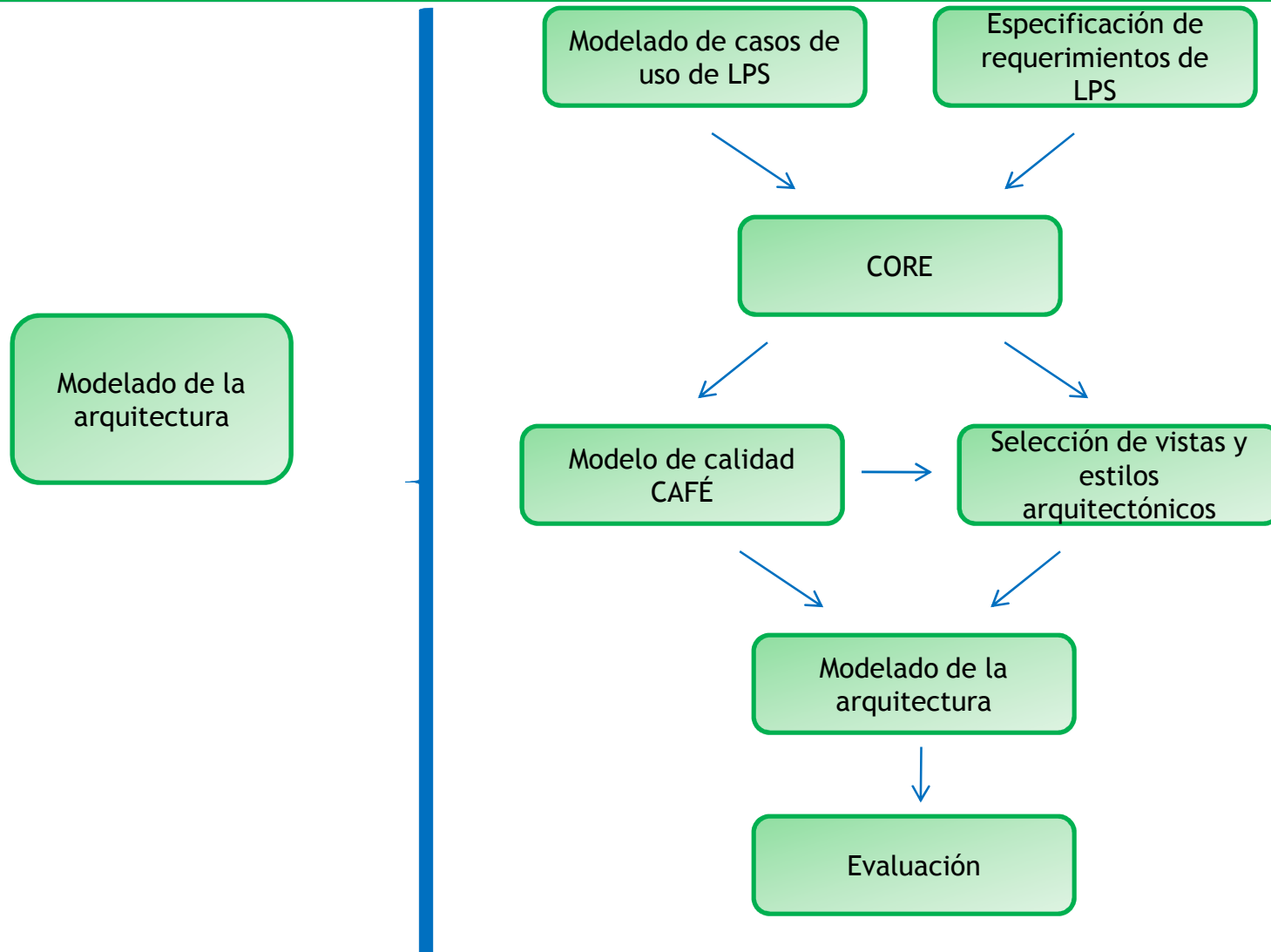
1. La identificación temprana de intereses (concerns)
2. *MDSOC (Multi-Dimensional Separation Of Concerns)*
3. La consideración de los intereses durante todo el proceso no sólo durante el modelado de la arquitectura



4. La definición de una PLA que incluye:
 - a) Atributos de calidad específicos de la LP,
 - b) Atributos de calidad de los productos
 - c) Soporte para las similitudes y variación,
 - d) Generalidad, y
 - e) Aspectos

PROCESO DE AOPLA

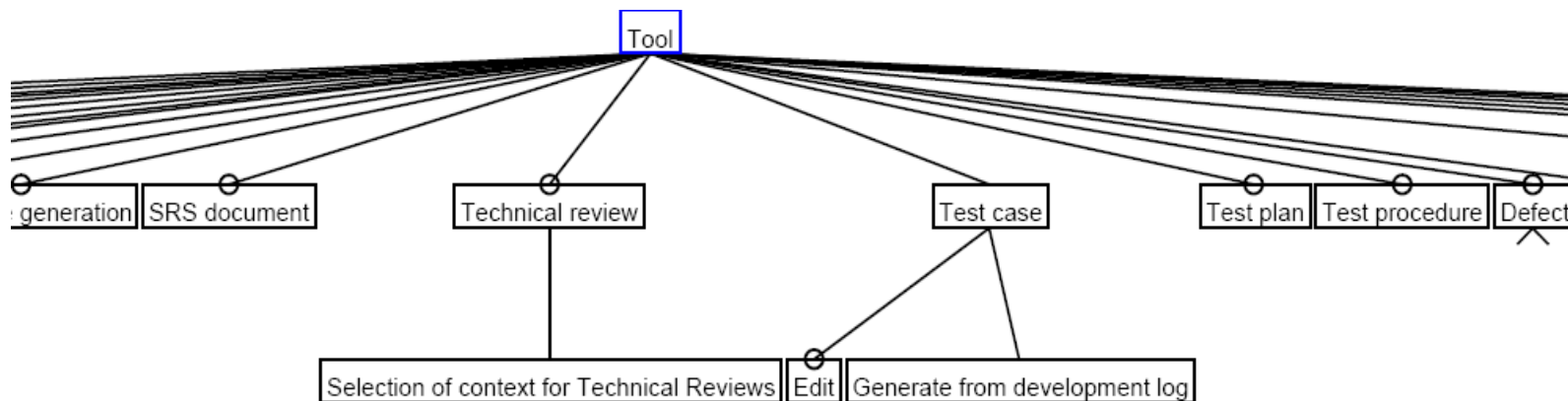






○ Análisis y modelado de dominios

- Modelo de contexto
- DER
- DFD
- Modelado de características





◉ Intereses lógicos

◉ Kinds

◉ **Functionality**

- ◉ Project
- ◉ Script
- ◉ Track
- ◉ Scene
- ◉ Quintet
- ◉ Dialogue
- ◉ Glossary

...

◉ **Entity**

- ◉ Project
- ◉ Script
- ◉ Track
- ◉ Scene
- ◉ Quintet

...

◉ **Feature**

- ◉ Function Points
- ◉ Use Case Points
- ◉ NSN
- ◉ Attitude Survey
- ◉ Glossaries and tables
- ◉ ERM
- ◉ ...



- **Property**
 - Evolvability
 - Generality
 - Derivability
 - Reusability
 - Security
 - Modifiability
 - Portability



- **Topics:**

- Function Points
- Use Case Points
- NSN
- Attitude Survey
- Glossaries and tables
- Script management
- Development Log

. . .

- Debido a su relación crosscutting:
 - Evolvability
 - Generality
 - Derivability
 - Reusability
 - Security
 - Modifiability
 - Portability



○ Intereses físicos

○ Instances:

- Context diagram
- Structure diagram
- Requirement
- Feature
- Architectural View

○ Collections:

- Requirements specification
- Feature model
- Architecture model



functionality is significant-for feature

Functionality	Feature
Cost.Function Points calculation	Function Points
Cost. Use Case Point calculation	Use Case Points

entity is significant-for feature

Entity	Feature
Cost.Function Point	Function Points
Cost. Use Case Point	Use Case Points
Test case	Test case.Edit
Test case	Test case.Generate from DL
Test plan	Test plan
Test procedure	Test procedure



○ Concern-Oriented Requirements Engineering

- Los intereses modelados con Cosmos se mapean a requerimientos y casos de uso
- La composición de intereses se guía con el portafolio de la LP
- Se mapean los intereses a componentes, decisiones arquitectónicas y aspectos (dimensiones)



Maps-to relationship (feature maps-to functionality maps-to requirement (or use case)

Feature	Functionality	Requirement (or use case)
Function Points.Selection of context	Cost.Function Points.Select Context	FPSelectionOfContextExtension_UC
Function Points.Calculation	Cost.Function Points calculation	FPCalculate_UC
Use Case Points.Selection of Context	Cost.Use Case Points.Select Context	UCPSelectionOfContextExtension_UC
Use Case Points.Calculation	Cost. Use Case Point calculation	UCPCalculate_UC



○ Mapeo de intereses

Concern	Influencia	Mapeo
Function Points	spec, design, impl	Component
NSN	spec, design, impl	Component
Glossaries and tables	spec, design, impl	Component
Development log	spec, design, impl	Component
Script management	spec, design, impl	Component
Technical reviews	spec, design, impl	Component
Test cases	spec, design, impl	Component
Defects	spec, design, impl	Component
Risk management	spec, design, impl	Component
MS-Office Compatibility	spec, design, impl	Component
Printing	spec, design, impl	Component
Project	spec, design, impl	Component
Portability	Arch, design, impl	Decision
Export	spec, design, impl	Aspect
Data dictionary	spec, design, impl	Component



○ Modelo de calidad de la LP

- Requerimientos de calidad: escenarios
- Métricas
- Patrones arquitectónicos
- Uso de medios



○ Atributos de calidad específicos de la LP

- F. de variación
- F. de derivación
- F. de reutilización
- F. de cálculo de valor (*rateability*)
- F. de integración
- F. de evolución
- F. de evolución
- F. de administración
- F. de mantenimiento

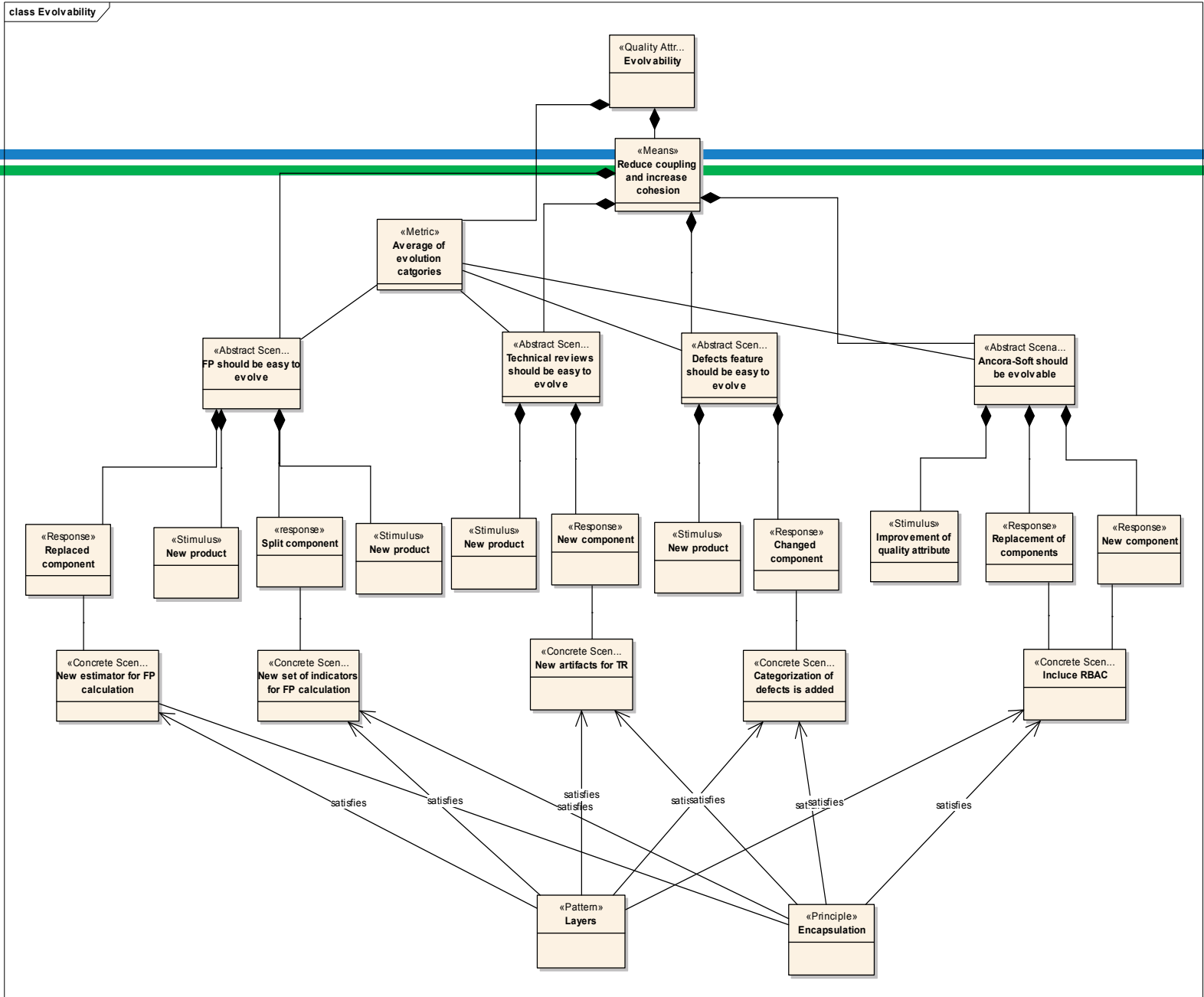


- Relación *property applies-to* o *kind (o instance) applies-to* requirement *applies-to* module *applies-to* component.

Property	Kind or instance	Requirement (or use case)	Module(s)	Component(s)
Evolvability	Entity.*	ALL		
Evolvability	Functionality.*	ALL	ALL	ALL
Evolvability	Features.*	ALL	ALL	ALL



- Vistas y estilos arquitectónicos
 - Modelo de calidad (PLQM)





- Vistas y estilos arquitectónicos
 - *Views and Beyond* (SEI)

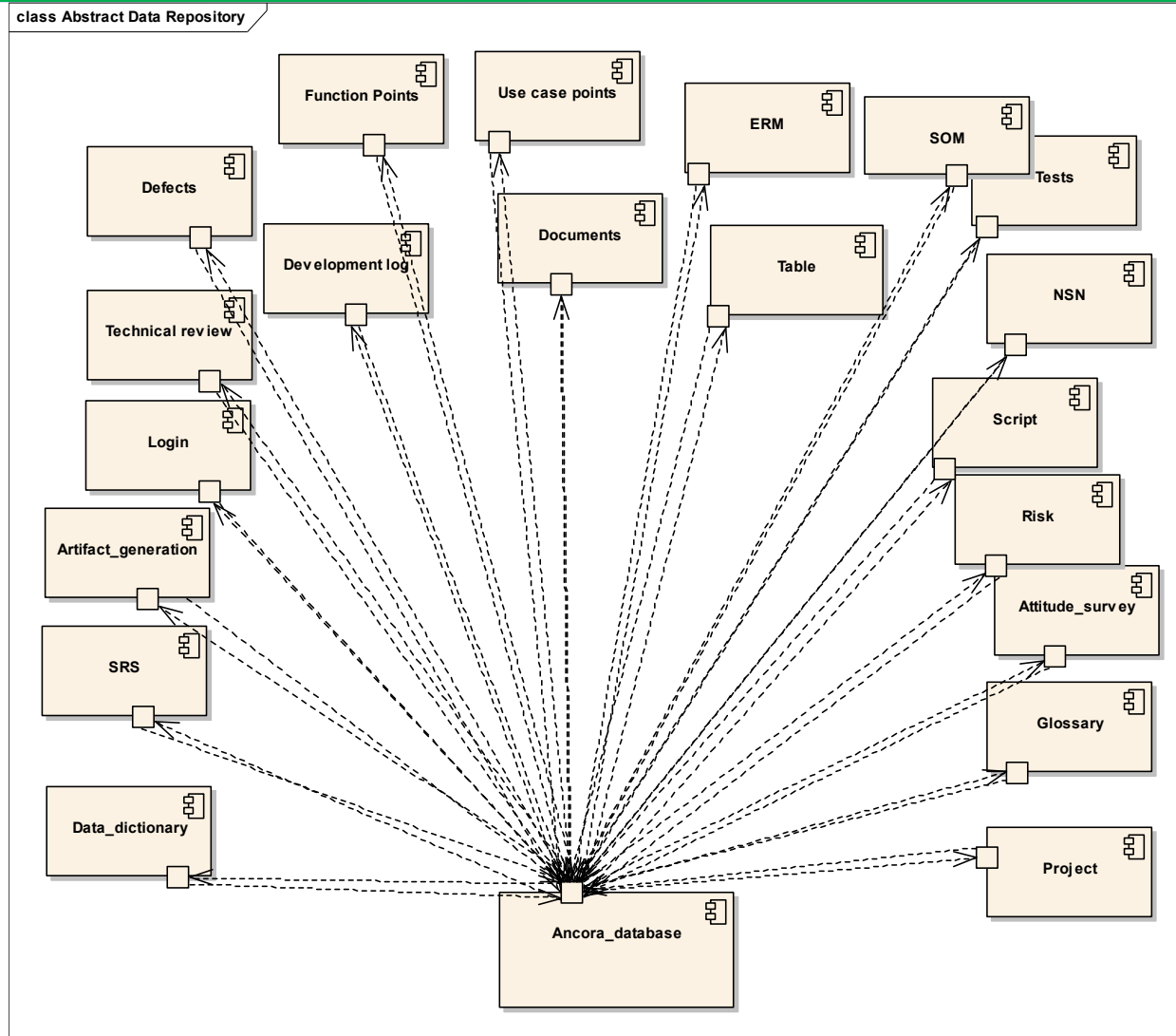
Atributo de calidad	Vista	Tipo de vista
F. de reutilización, F. de derivación, generalidad	Generalización, usa	Módulos
F. de modificación	Descomposición capas	Módulos
F. de evolución	Descomposición, Capas, Generalización	Módulos
Portabilidad	Capas	Módulos
Seguridad	Despliegue	Asignación

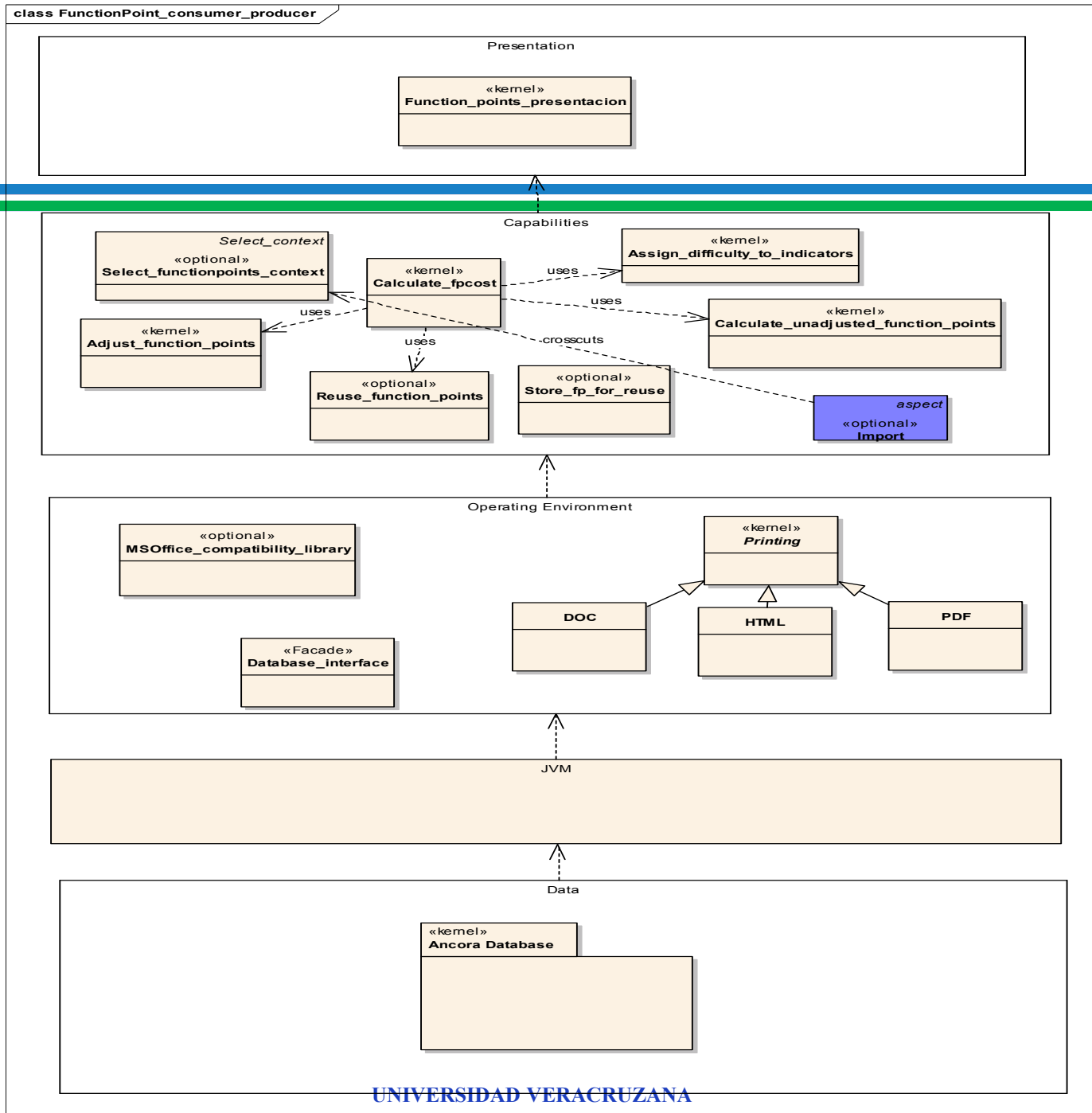


- Modelado de la arquitectura
 - Durante el desarrollo de los tipos de vista (*viewtypes*), se incorporan los patrones, principios, técnicas y tácticas del modelo de calidad
 - Se consideran las dimensiones de CORE
 - Se modelan las similitudes y la variación
 - La derivación de las arquitecturas de los productos se hace en el tipo de vista de C&C
 - Se incorpora una vista con aspectos



Abstract Data Repository ABAS







○ Mapeo de intereses y administración de intereses

Características	Funcionalidad	Requirimiento (o caso de uso)	Módulo(s)	Componente(s)
Function Points.Selection of context	Cost.Function Points.Select Context	FPSelectionOfContextExtension_UC	Select_functionpoints_context	Select_functionpoints_Context
Function Points.Calculation	Cost.Function Points calculation	FPCalculate_UC	Assign_difficulty_to_indicators	Indicators_Difficulty
			Calculate_unadjusted_function_points	Unadjusted_FunctionPoints
			Adjust_function_points	Adjust_FunctionPoints
			Calculate_fpcost	Calculate_FunctionPoints
Use Case Points.Selection of Context	Cost.Use Case Points.Select Context	UCPSelectionOfContextExtension_UC	Select_usecasepoints_context	Select_UseCasePoints_Context
Use Case Points.Calculation	Cost. Use Case Point calculation	UCPCalculate_UC	Assign_difficulty_to_indicators	UseCasePoints_Indicators_Difficulty
			Calculate_unadjusted_usecase_points	Unadjusted_UseCasePoints
			Adjust_usecase_points	Adjust_UseCasePoints
			Calculate_usecase_cost	Calculate_UseCasePoints
NSN	NSN.Create	NSN_UC	Collect_concepts	Collect_NSN_Concepts

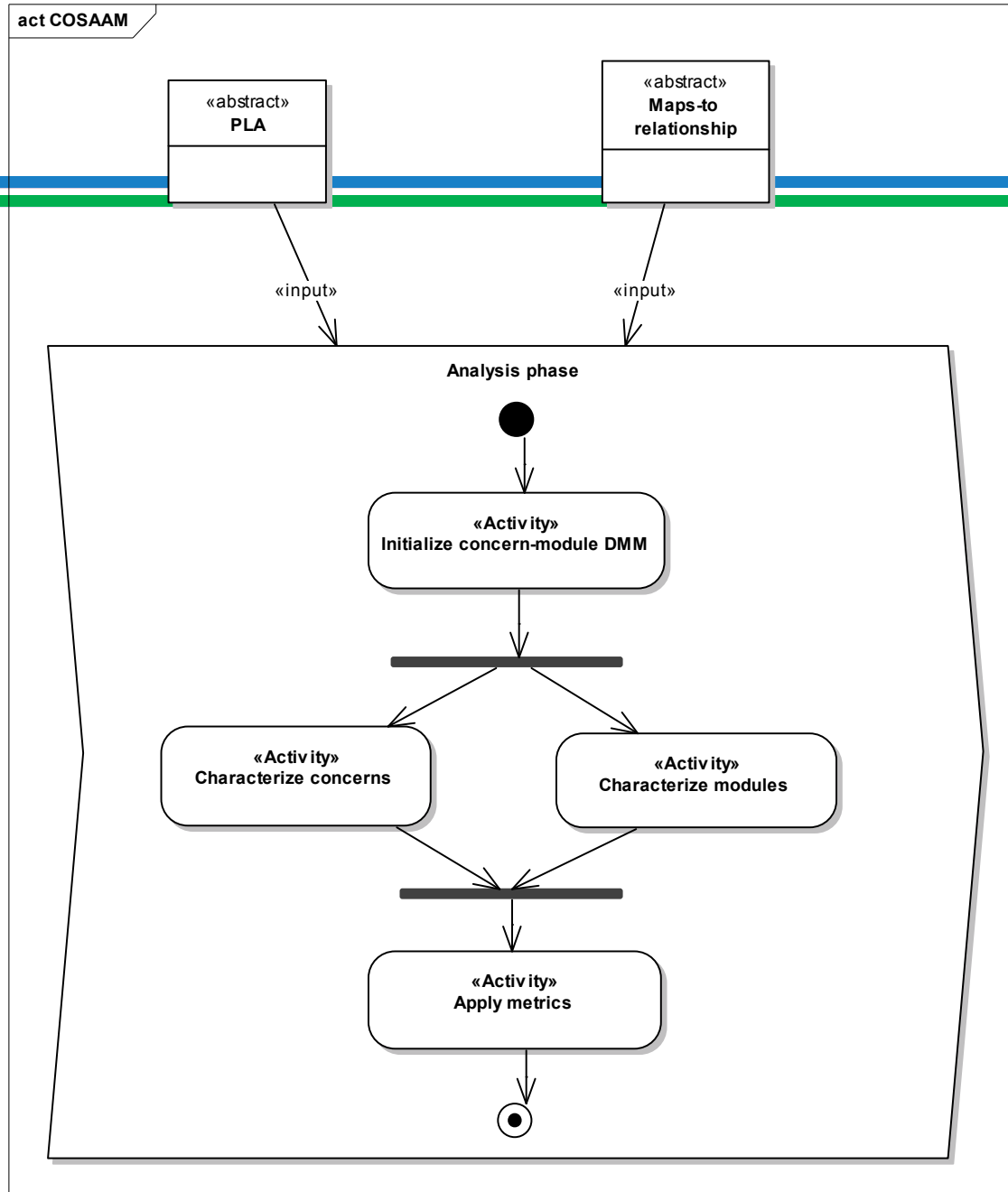


- Evaluación
 - Métricas del PLQM
 - ATAM (Architecture Tradeoff Analysis Method)
 - COSAAM (Concern-Oriented Software Architecture Analysis Method)



◎ COSAAM

- Fase de preparación:
 1. Define los intereses
 2. Describe la arquitectura candidata
- Fase de análisis:
 1. Inicializar la *concern-module DMM (Domain Mapping Matrix)*
 2. Caracterizar el mapeo de intereses y módulos
 3. Medir *Scattering yTangling*
- Fase de transformación:
 1. Inicializar y secuenciar la DSM (Desing Structure Matrix) de la arquitectura
 2. Seleccionar la regla de transformación
 3. Aplicar la regla de transformación





- Como resultado de la medición de *scattering* y *tangling* se obtuvo que:
 - Los módulos que se obtienen durante el proceso son altamente cohesivos
 - El máximo grado de *tangling* es muy bajo (en un mismo módulo hasta 3 intereses de un máximo de 52)
 - El máximo grado de *scattering* (34) fue resuelto como un aspecto durante el diseño de la arquitectura

CONCLUSIONES

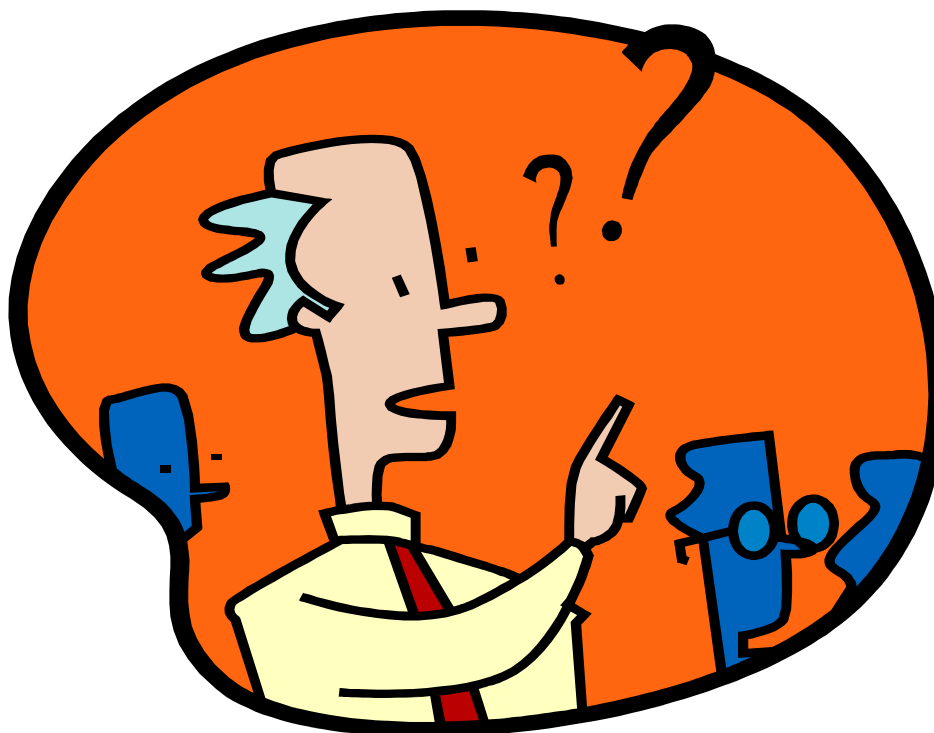


- AOPLA es un método para el diseño de PLAs que tiene un enfoque orientado a intereses temprano que permite:
 - la identificación y modelado de intereses a lo largo del proceso
 - la consideración de intereses en la PLA (componentes, decisiones arquitectónicas y aspectos)
 - el desarrollo del modelo de calidad
 - módulos altamente cohesivos

¿PREGUNTAS?



Universidad Veracruzana



cortesverdin@prodigy.net.mx

kcortes@uv.mx