



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA



Departamento de Informática
Universidad Técnica Federico Santa María

Ingeniería de Software

Feature Driven Development

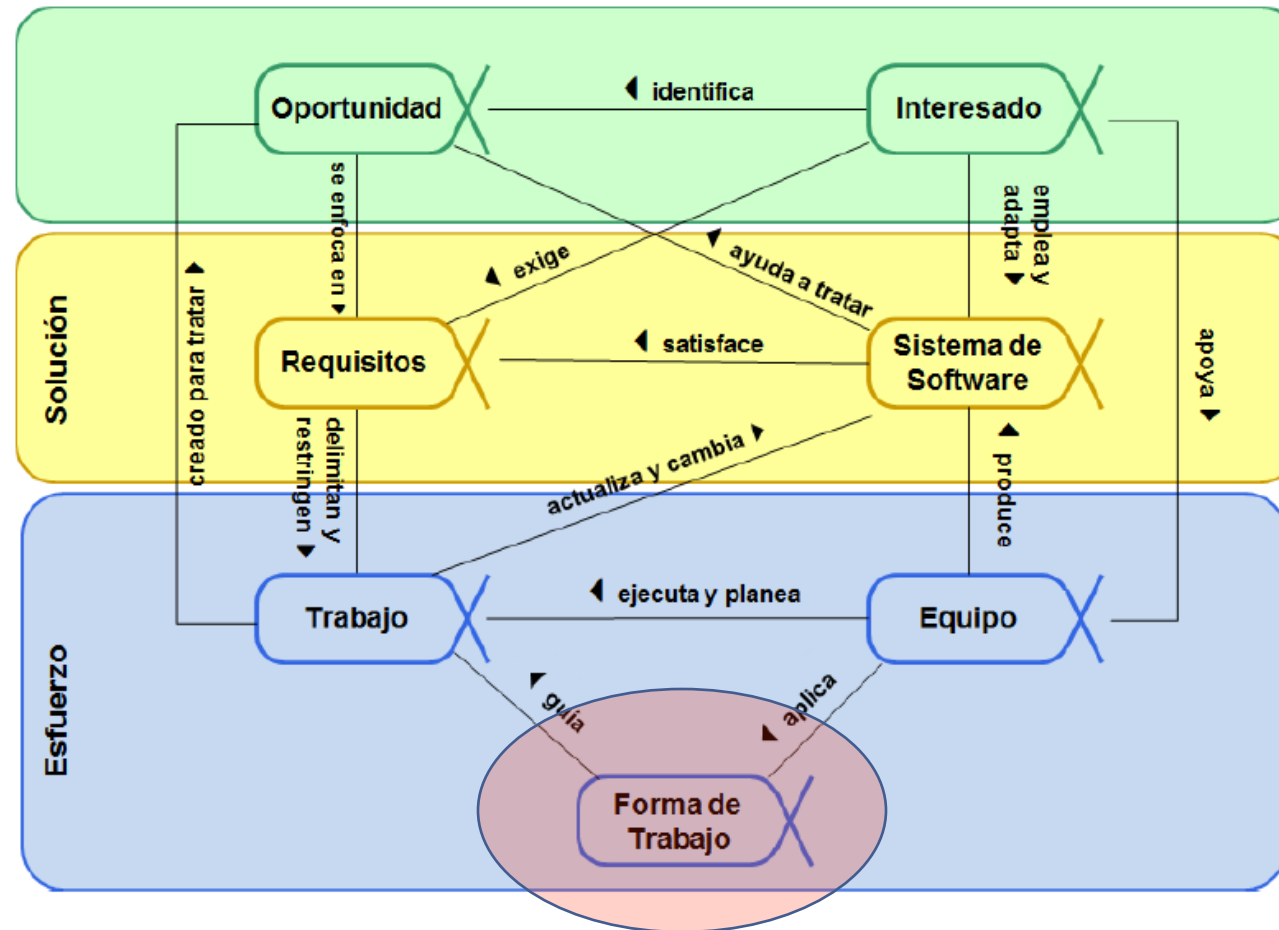
Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María

Contexto

Contexto

- Como se ha visto en las sesiones anteriores, el modelo de cascada se divide en diferentes fases
 - Se asume que en cada etapa todo debe estar 100% listo para poder pasar a la siguiente
 - Lo anterior, produce complicaciones en el modelo
- Las metodologías ágiles intentan evitar esta debilidad
 - Iteraciones incrementales ayudan
- FDD nace como alternativa al desarrollo en cascada tradicional

Contexto en SEMAT



Feature Driven Development

FDD [1]

- ¿Qué es FDD?
 - Es un proceso de desarrollo ágil y altamente adaptativo con las siguientes cualidades
 - Iteraciones cortas
 - Enfatiza la calidad en todos los pasos
 - Entrega resultados frecuentes, tangibles en todos los pasos
 - Provee progresos e información en cada estado, con el mínimo de problemas para los desarrolladores
 - Es amigable para clientes, gerentes y desarrolladores

FDD [2]

- Problemas tradicionales en el desarrollo de software
 - Complejidad→ Se sabe que el proyecto es complejo en varios aspectos, pero siempre existe el problema para estimar
 - FDD→ Descompone el dominio del problema en pequeños problemas, los cuales pueden ser resueltos en pequeños periodos de tiempo, usualmente 2 semanas.
 - FDD→ Divide el proyecto en iteraciones para que la distancia de tiempo entre el análisis y el testing sea poco, así se descubren errores tempranamente.
 - Calidad→ En el desarrollo de software regularmente se pide productos de calidad, pero no se sabe cómo manejarla durante el proceso de desarrollo
 - FDD→ El concepto de calidad es abordado no solamente en el test del código, sino también la incluye en estándares de código, mediciones y métricas.

FDD [3]

- Roles en FDD
 1. Project Manager
 2. Chief Architect
 3. Development Manager
 4. Chief Programmers [progr. principal; construye subsistema; “cirujano”]
 5. Class Owners (más conocidos como desarrolladores)
 6. Domain Experts

Roles de FDD

Roles de FDD [1]

- **Project Manager**
 - Gestiona y lidera los objetivos del proyecto
 - Presupuesto, personal, progreso en los reportes
 - Opera el sistema del proyecto
 - Maneja posibles distracciones del personal
- **Chief Architect**
 - Es responsable del diseño del sistema
 - Ejecuta talleres de diseño
 - Prepara al equipo para superar obstáculos técnicos

Roles de FDD [2]

- **Development Manager**

- Lidera el día a día de las actividades de desarrollo
- Resuelve los conflictos de recursos
- Se apoya entre el Chief Architect y Project Manager

- **Chief Programmers**

- Desarrolladores con experiencia
- Lideran pequeños grupos de desarrolladores
- Rol principal: maneja las necesidades entre desarrolladores y managers

Roles de FDD [3]

- Class Owners
 - Desarrolladores individuales
 - Diseño, test, código y documentación
- Domain Experts
 - Usuarios, clientes, partners, sponsors, otros.
 - Expertos en el conocimiento

Roles de FDD [4]

- Roles de soporte para el equipo
 - Supporting roles
 - Domain Manager
 - Release Manager
 - Language guru
 - Build Engineer
 - Toolsmith
 - System Administrator
 - Tester
 - Deployers
 - Technical writers

Prácticas en FDD

Prácticas en FDD [1]

- FDD aplica prácticas que son aceptadas en la industria las cuales son:
 1. Domain Object Modeling
 2. Developing by Feature
 3. Class (Code) Ownership
 4. Feature team
 5. Inspections
 6. Regular Build Schedule
 7. Configuration Management
 8. Progress Reporting

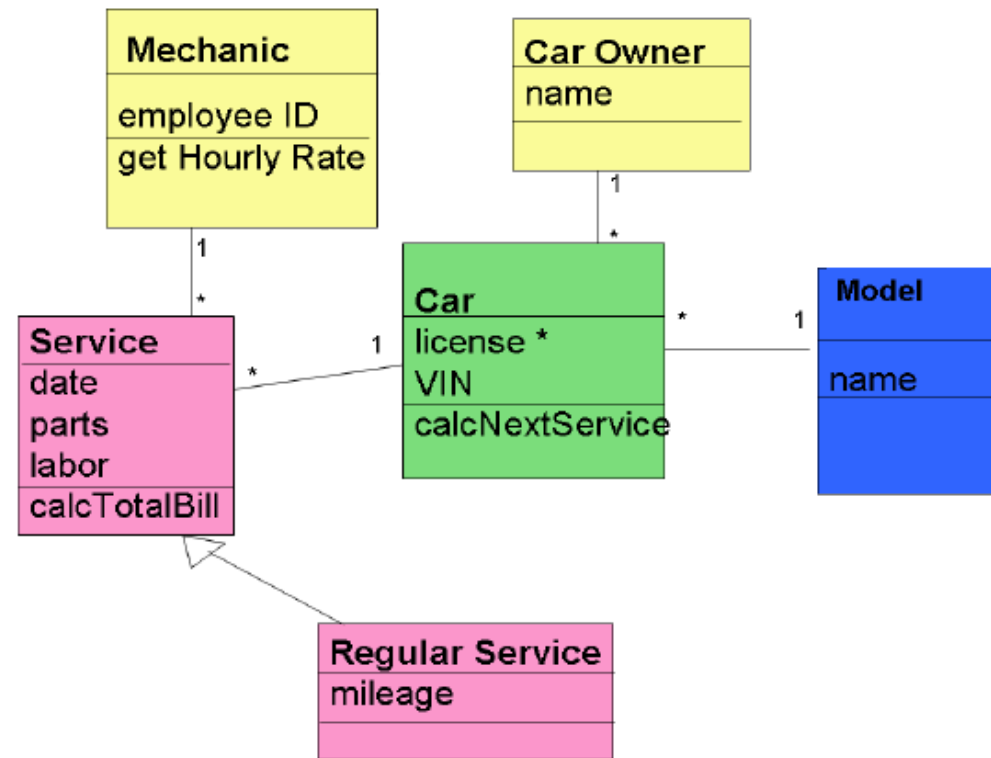
Prácticas en FDD [2]

- **Domain Object Modeling**

- Construcción de diagramas de clases en función de su significado
- Es una forma de descomposición de objetos
- Modelar con colores [Goyal, 2007]
 - Amarillo → rol utilizado por personas u organizaciones
 - Azul → descripciones de catalogo
 - Verde → lugar o cosa
 - Rosado → momento o intervalo de tiempo

Prácticas en FDD [3]

- Domain Object Modeling



Prácticas en FDD [4]

- **Developing by Feature**

- Se entiende como *feature* un característica pequeña que puede ser implementada en 2 semanas.
- El template de un feature es
 - <action>the <result><by|for |of | to|><a(n)><object>
 - Por ejemplo: Calcular[action] el total[result] de la venta[object]
- Un *feature set* es un grupo de características del negocio
 - <action><-ing><a(n)><object>
 - Por ejemplo: Making a product sale (Creando una venta de producto)

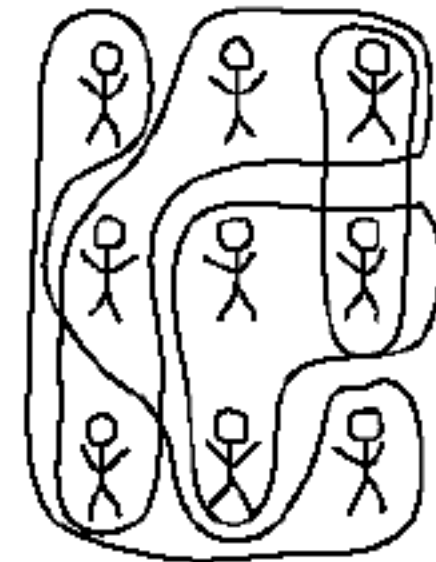
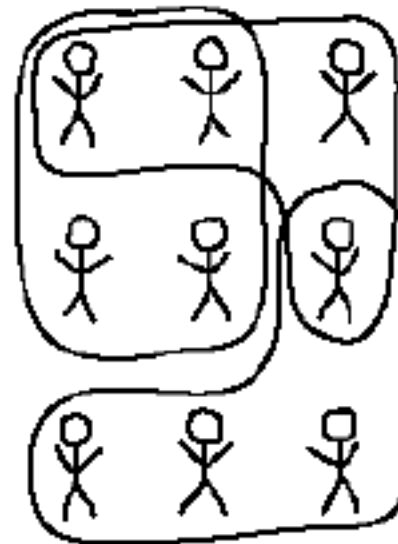
Prácticas en FDD [5]

- **Class (Code) Ownership**

- Denota quién o que rol es responsable del contenido de la clase
- Se asignan responsabilidades sobre la codificación de esa clase

- **Feature team**

- Conformado por equipos pequeños
- Capacidad de poder cambiarse de roles



Prácticas en FDD [6]

- **Inspections**

- Para evitar problemas, los Features Team inspeccionan todas las nuevas dimensiones del proyecto
- Se aplican técnicas de detección de errores
- Se buscan nuevas oportunidades

- **Regular Build Schedule**

- Se revisa que que todos los intervalos de código definidos por el equipo se vayan complementando
- Esto permite que siempre habrá un sistema demostrable

Prácticas en FDD [7]

- **Configuration Management**

- Se gestionan las últimas versiones de código
- Objetivo → crear un historial sobre seguimiento de los artefactos del proyecto

- **Progress Reporting**

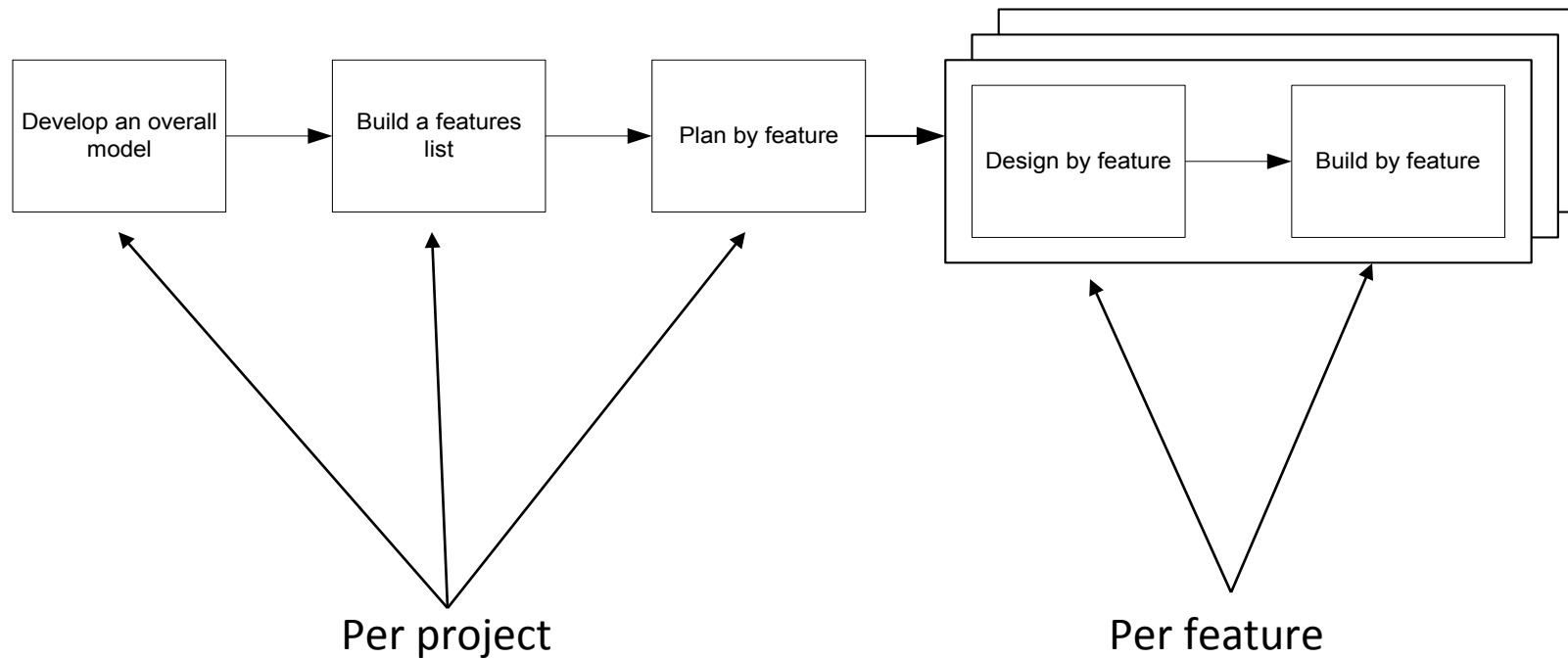
- Apropiaada documentación en todos los niveles
- Respaldo de trabajo terminado

Proceso de FDD

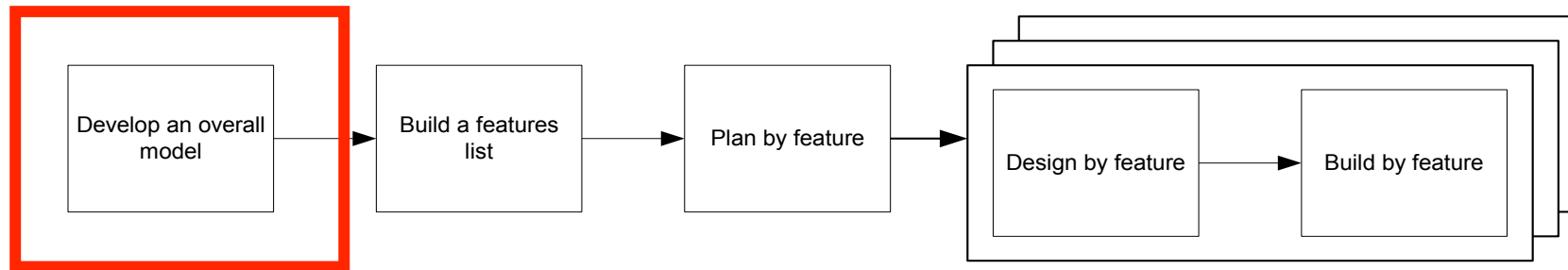
Proceso de FDD [1]

- FDD empieza con la creación del Modelo de Dominio de Objeto con la colaboración de Domain Experts.
- Usando la información obtenida, los desarrolladores crean una lista de características.
- Luego, el plan es dibujado y las responsabilidades son asignadas.
- Se conforman pequeños grupos en función de las características en donde el desarrollo de cada característica no pase de 2 semanas.
- En la siguiente slide se presenta el proceso [Goyal, 2007][Calberg, 2007]

Proceso de FDD [2]

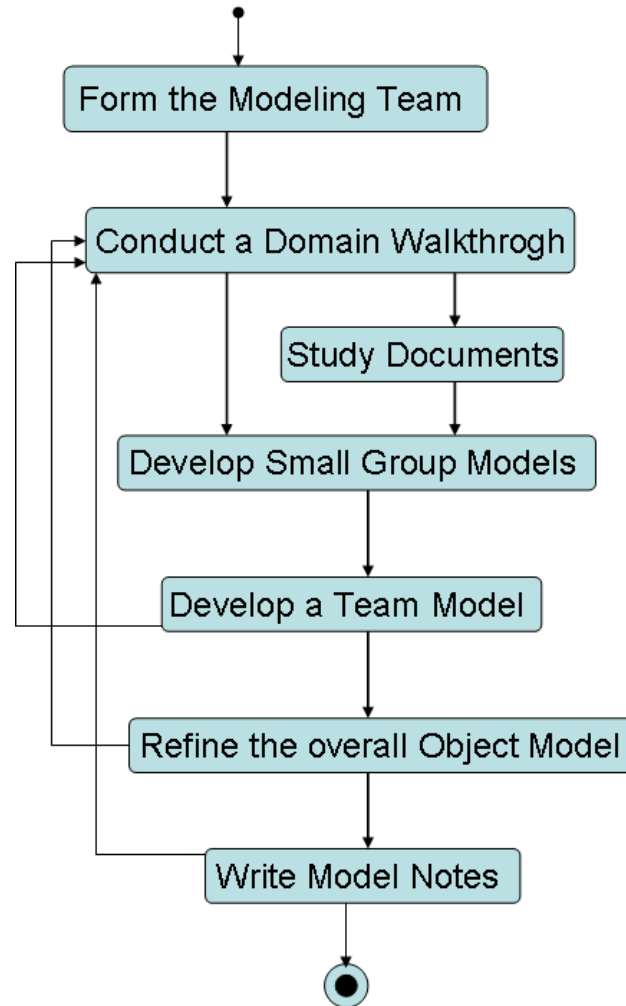


Proceso de FDD [3]



Proceso de FDD [4]

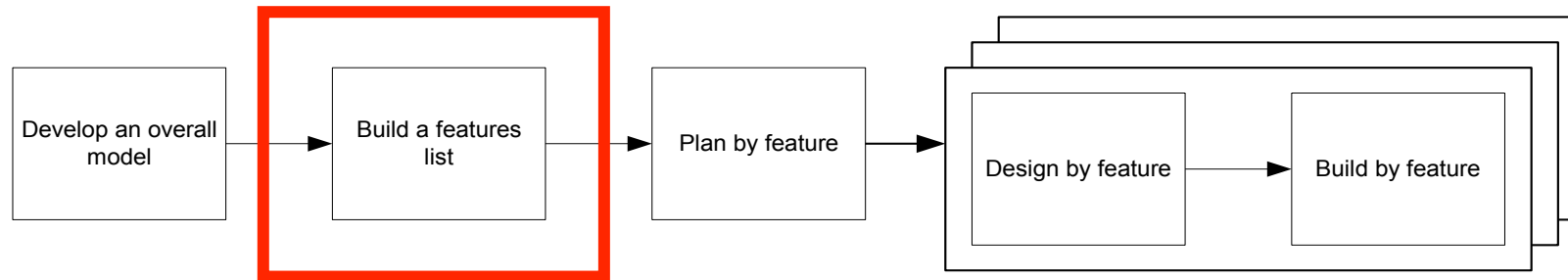
- [Palmer et al., 2002]



Proceso de FDD [5]

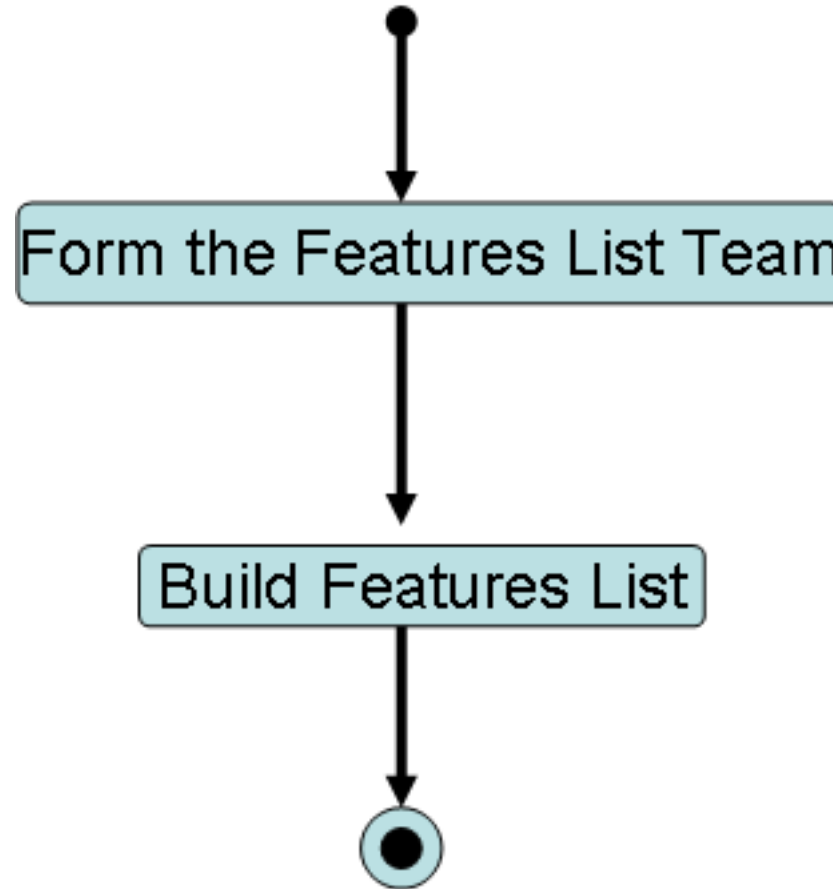
- **Develop an overall model**
 - ¿Quién participa?
 - Domain Experts, Chief Architect, Chief Programmers
 - Se establece la forma del proyecto
 - Se definen las clases y las relaciones
 - Se crea la base del Modelo de Objetos
 - Se incluyen entrevistas con el cliente

Proceso de FDD [6]



Proceso de FDD [7]

- [Palmer et al., 2002]

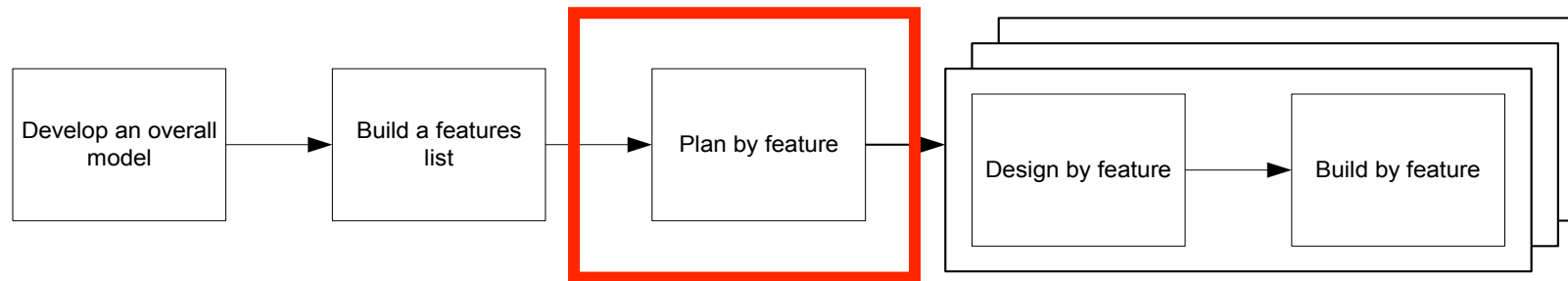


Proceso de FDD [8]

- **Build a features list**

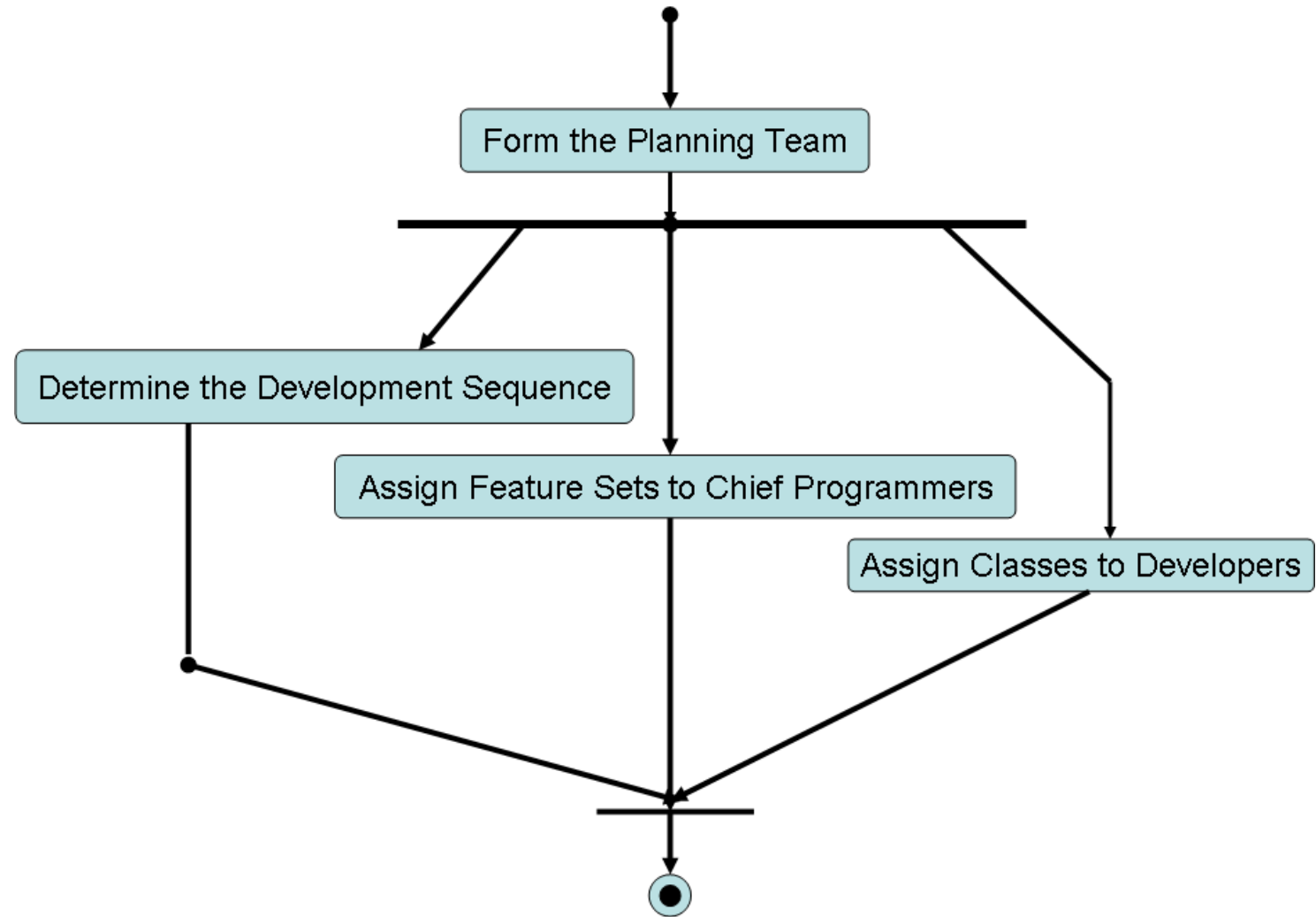
- ¿Quién participa?
 - Domain Experts, Chief Programmers, Chief Architect
- Descomposición funcional del modelo
- Se analizan las tareas del negocio y se traducen en tareas de equipo
- Se comienza a generar valor al cliente

Proceso de FDD [9]



Proceso de FDD [10]

- [Palmer et al., 2002]

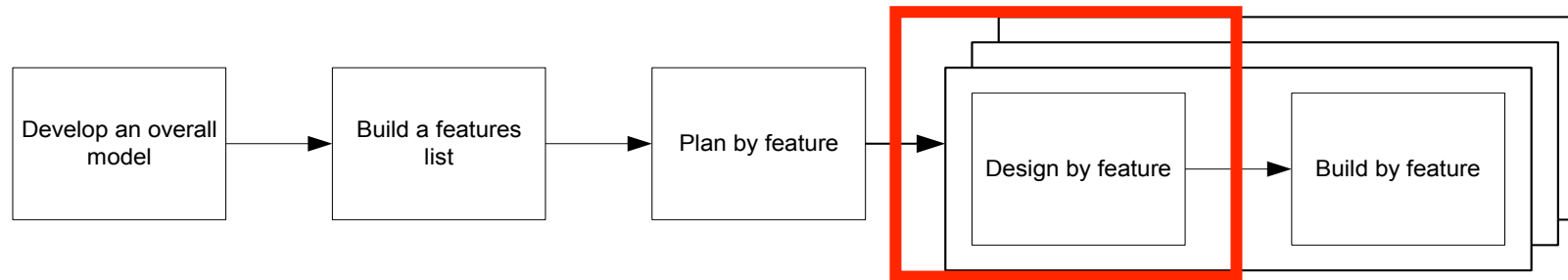


Proceso de FDD [11]

- **Plan By Feature**

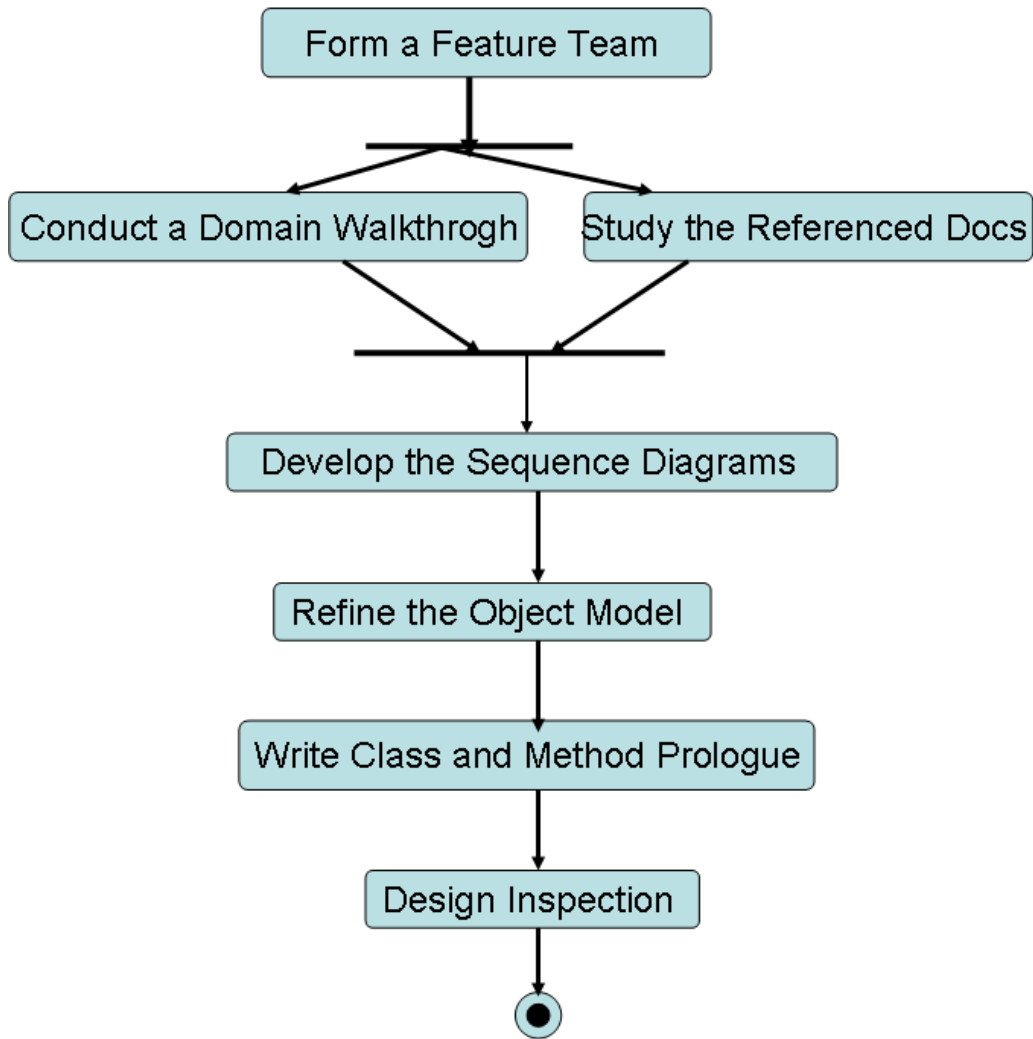
- ¿Quién participa?
 - Project Manager, Development Manager y Chief Programmers
- Se clasifica el grupo de características en conjuntos de características
- Se prioriza en base a las necesidades del cliente
- Se establecen fechas de compromisos

Proceso de FDD [12]



Proceso de FDD [13]

- [Palmer et al., 2002]

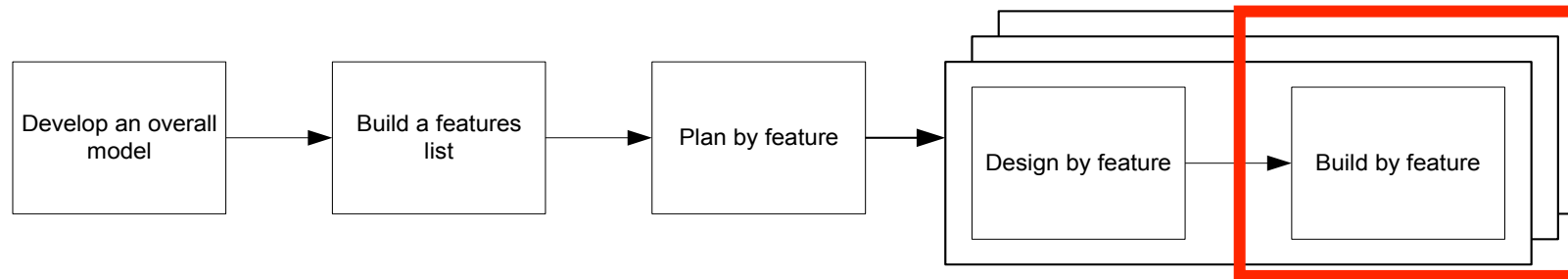


Proceso de FDD [14]

- **Design by feature**

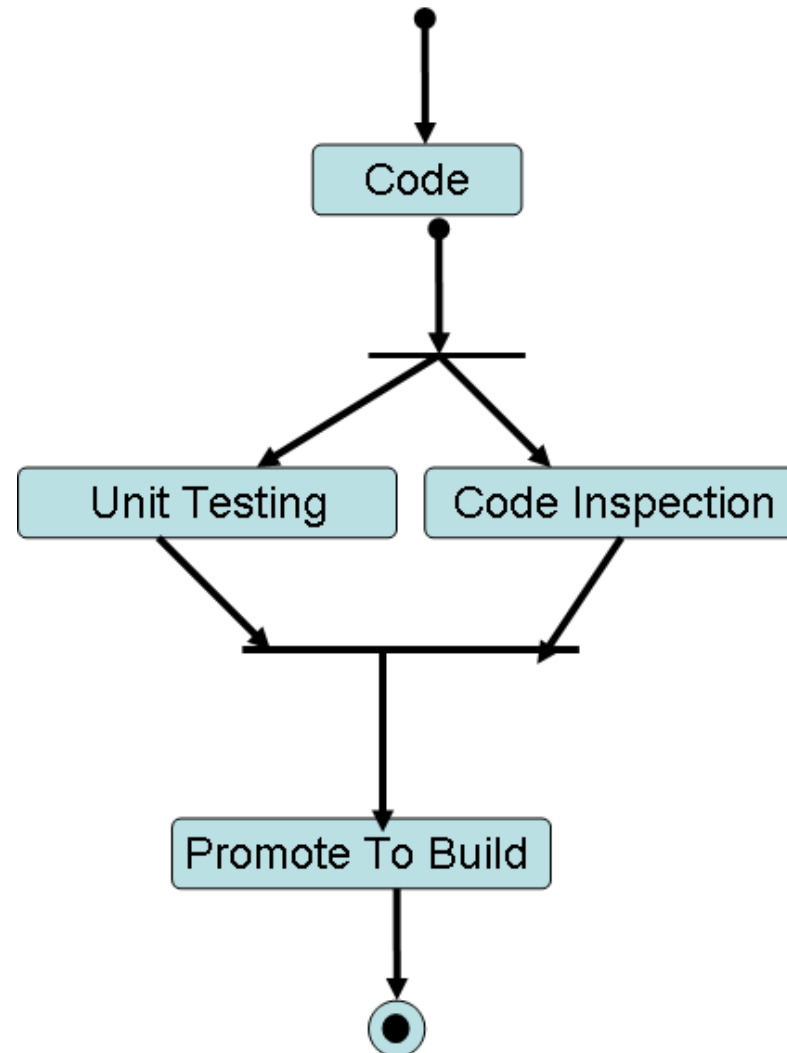
- ¿Quién participa?
 - Chief Programmer, Class Owners
- Se analiza la arquitectura técnica del proyecto
- 2 semanas de trabajo
- Se crean diseño de clases y objetos, diagramas de secuencia (si es necesario)
- La fluidez de comunicación entre los equipos es primordial
- Se pueden hacer actualizaciones al Modelo de Objetos

Proceso de FDD [15]



Proceso de FDD [16]

- [Palmer et al., 2002]



Proceso de FDD [17]

- Develop by feature
 - ¿Quién participa?
 - Class owners, Chief Programmers
 - Implementar
 - Inspección de código
 - Pruebas unitarias
 - Promoción a producción

Conclusiones

Conclusiones [1]

- [Calberg, fivesticks.com]

	RUP	FDD	XP
Scales To	???	10-250 developers	50 developers
Tools	Rational	TogetherSoft (Borland)	???
Process	Heavy	Medium	Agile
Roles	~30	~6 (9 optional)	~7
Artifacts	25-30	Flexible ~10-15	~30

Conclusiones [2]

- FDD es un método que ayuda a equipos a producir trabajo tangible.
- Es útil para funcionalidad del cliente (features)
- FDD se enfoca que el equipo muestre resultados en 2 semanas y aporta seguimiento y reporte de capacidades
- Herramienta para trabajar con FDD
 - <http://borland.com/en-GB/Products/Requirements-Management/Together>



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA



Departamento de Informática
Universidad Técnica Federico Santa María

Ingeniería de Software

Feature Driven Development

Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María