



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA



Departamento de Informática  
Universidad Técnica Federico Santa María

# SEMAT

Ingeniería de Software – INF-225

**Hernán Astudillo & Gastón Márquez**

*Departamento de Informática*

*Universidad Técnica Federico Santa María*

<hernan@inf.utfsm.cl, g.marquez.o@gmail.com>

# Contexto

# Motivación [1]

- Existen millones de ingenieros de software en el planeta que trabajan en programas, proyectos y equipos
- Pero, aún es difícil compartir buenas prácticas
  - Integrar equipos de ingeniería de software a otras disciplinas
- Big Data, Cloud Computing y la industria del internet son recursos importantes en el desarrollo de software
- Tener habilidades y prácticas sofisticadas de ingeniería de software es primordial para el éxito.
- Se necesitan muchos tipos diferentes de prácticas más allá de las prácticas ágiles más comunes (Scrum, historias de usuarios, etc.) para tener éxito.
- Estos incluyen prácticas de arquitectura, sistema de sistemas, casos de uso, desarrollo orientado al servicio, equipos de equipos, mejora de procesos, etc.

# Motivación [2]

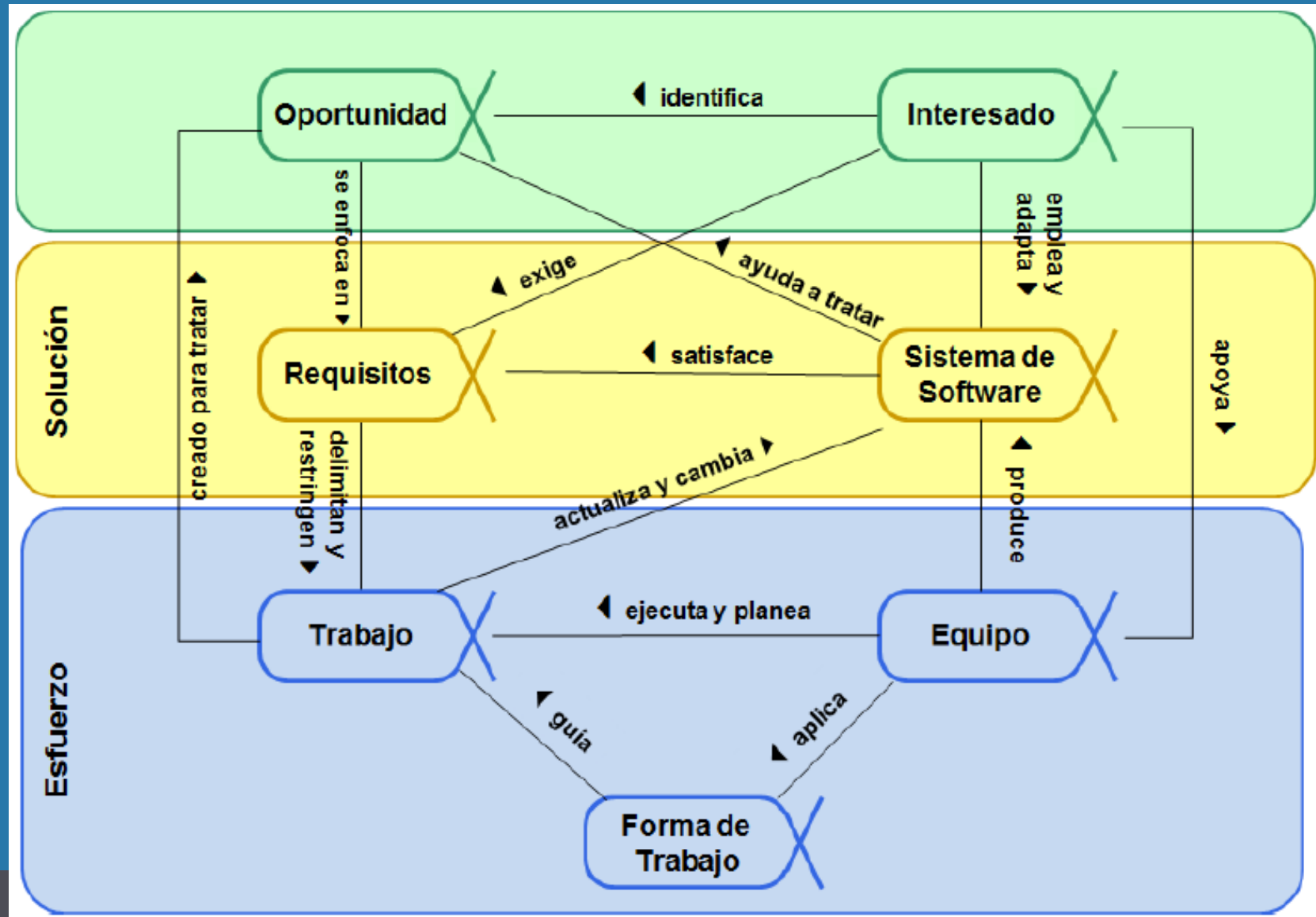
- Quienes construyen software saben que es un negocio riesgoso
- Afortunadamente, la Ingeniería de software es una disciplina joven
- El éxito del desarrollo de software depende de los siguientes factores:
  - Entrega rápida
  - Satisfacción de los stakeholders
  - Gestión de riesgos
  - Mejora constante

# Motivación [3]

- Escenario actual de la industria del software (Levison, 2010):
  - Se sigue mucho las modas
  - No hay una *teoría fundamental* del área
  - Muchos métodos son artificiales
  - Poca evaluación experimental
  - Poca conexión entre industria y academia

# SEMAT

# Figura motivadora



# ¿Qué es SEMAT? [1]

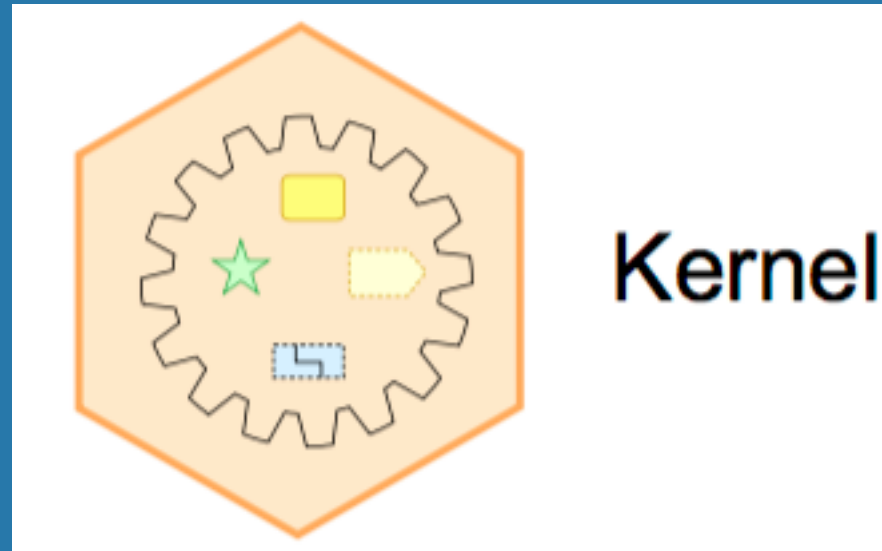
- SEMAT = Software Engineering Method And Theory
- Iniciativa para *reformular la base* de la Ingeniería de Software
  - Comunidad abierta de voluntarios
  - Iniciada en 2009 por Ivar Jacobson, Bertrand Meyer & Richard Soley
  - Ahora: 30 famosos + 15 instituciones + 1600+ “supporters”
- Propósito
  - Crear una teoría general de Ingeniería de Software
  - Desarrollar un Kernel común que incluya la esencia de la Ingeniería de Software



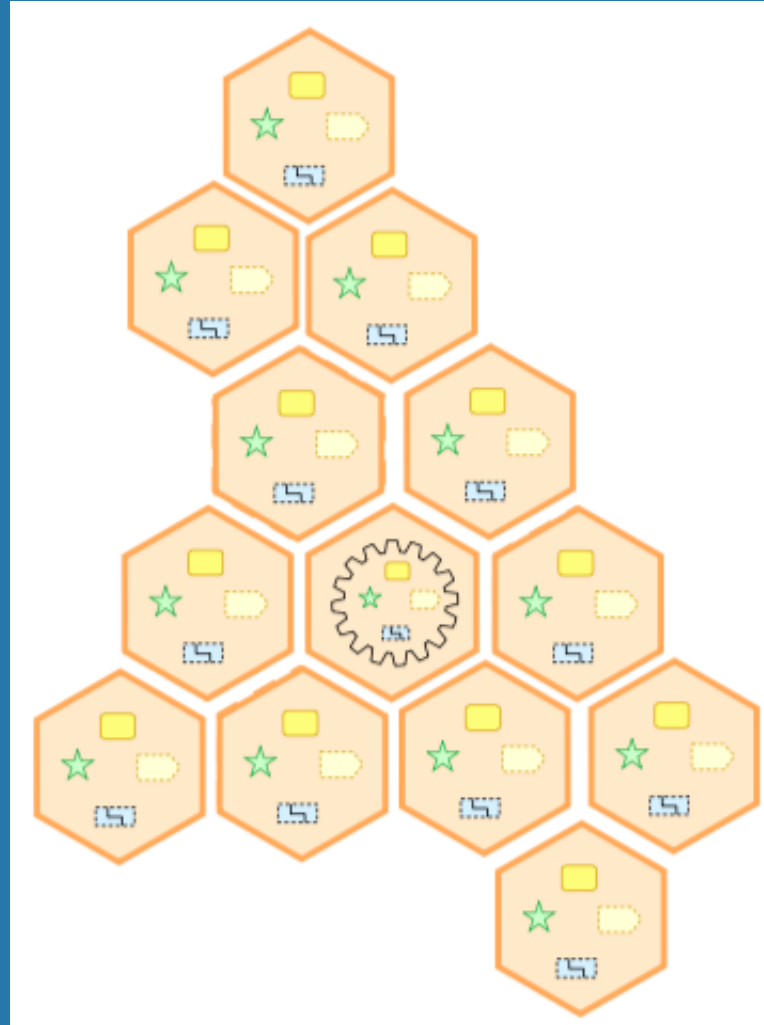
# Idea base: el "kernel"

- El *kernel* es lo que todo esfuerzo de desarrollo tiene en común
  - “Cosas con las que siempre trabajamos”: alfas
  - “Cosas que siempre hacemos”: prácticas
  - “Habilidades que siempre necesito saber”: tarjetas

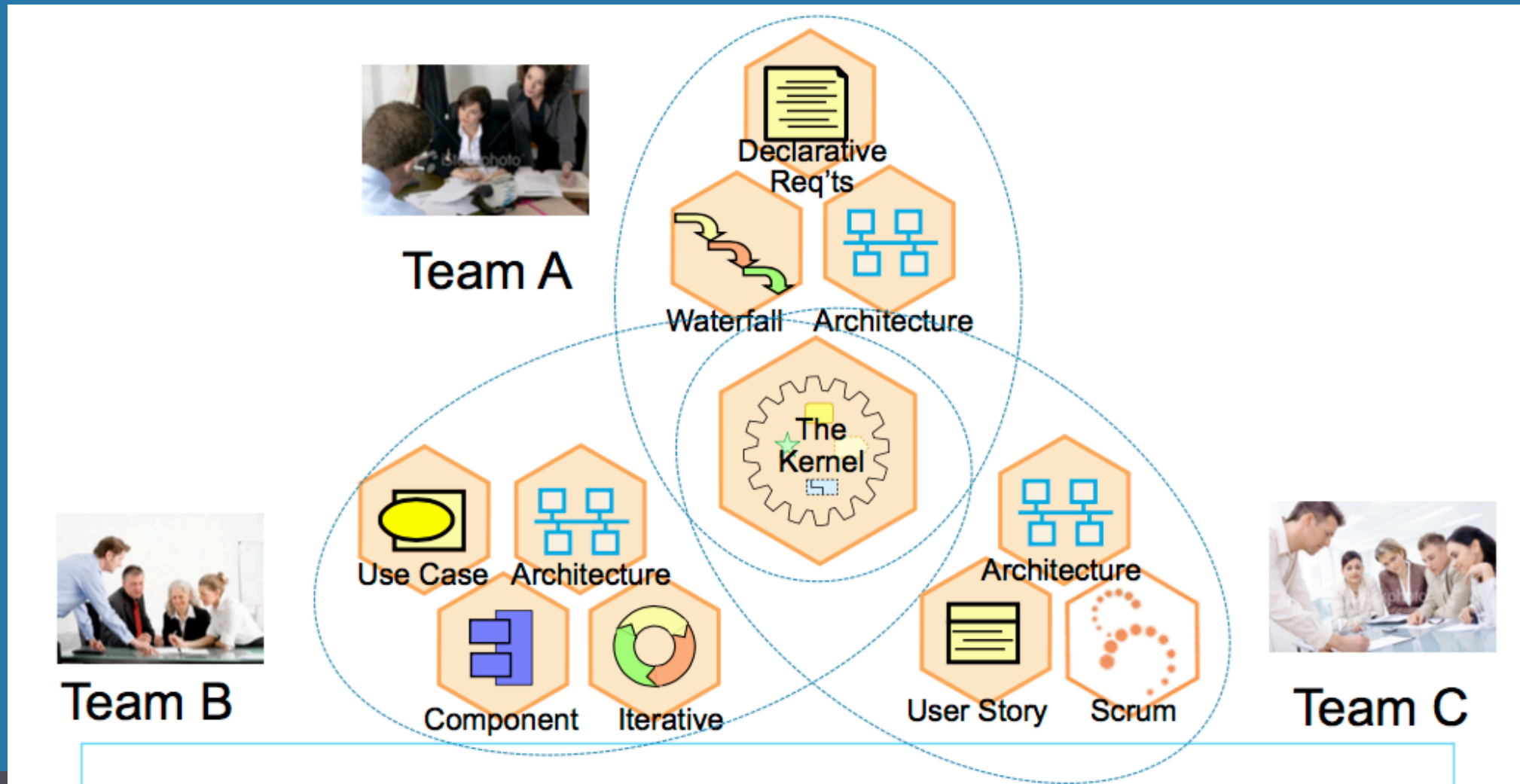
# Idea base: el "kernel"



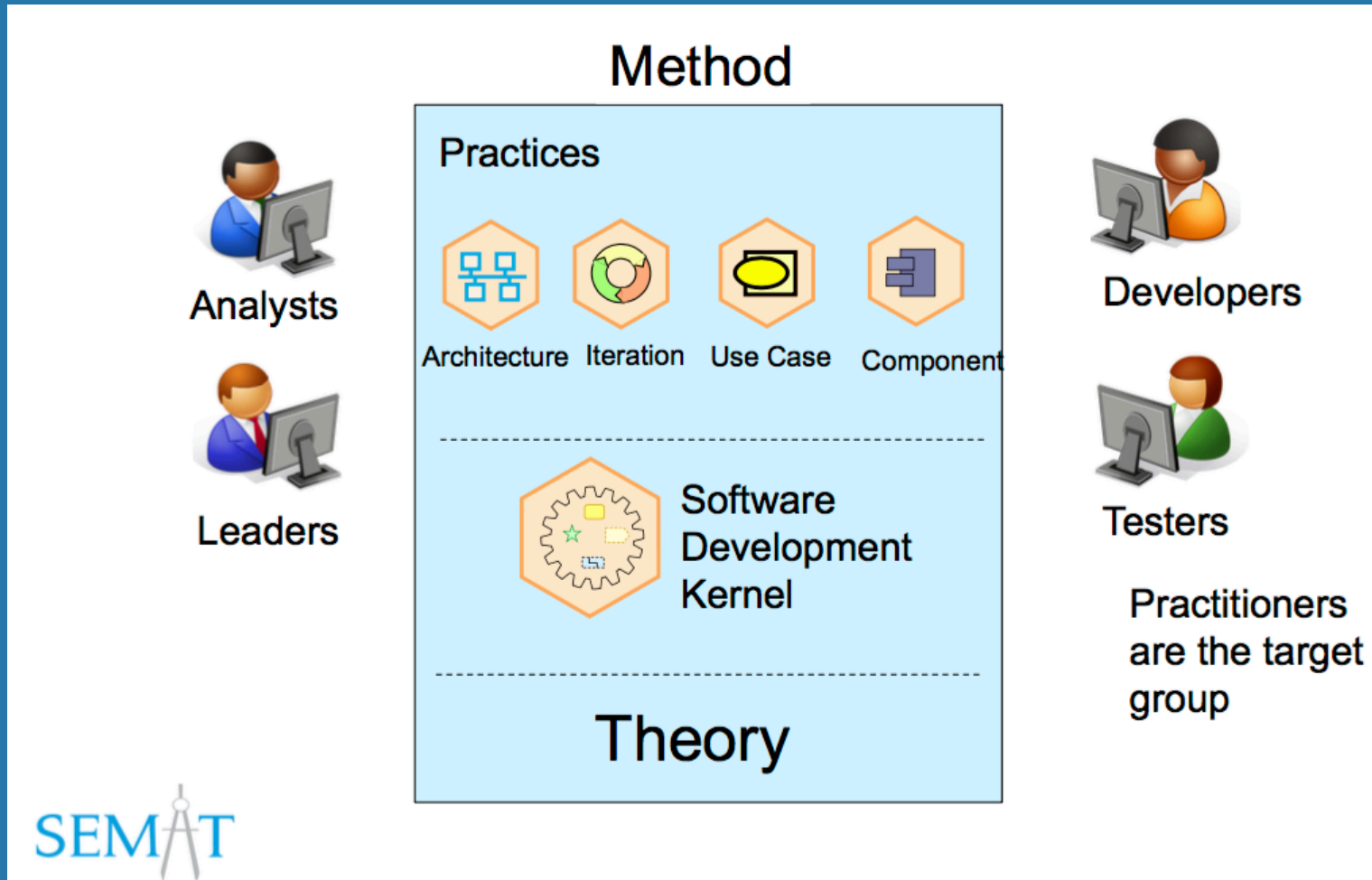
# Idea base: el "kernel"



# Idea base: el "kernel"

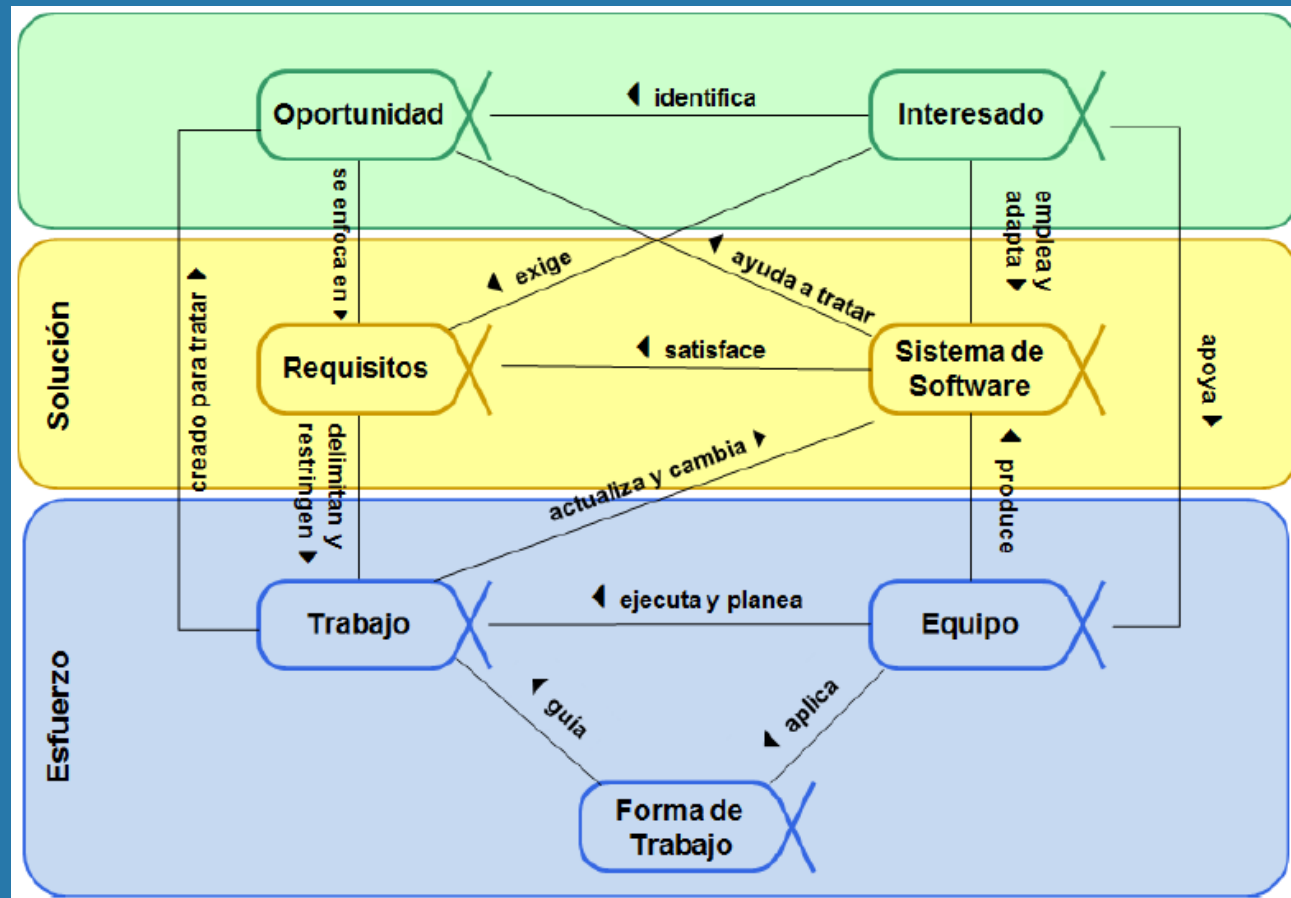


# Idea base: el "kernel"



# Kernel: alfas [1]

- “Cosas con las que siempre trabajamos”



# Kernel: alfas [2]

- *alfas*: “cosas con que siempre trabajamos”
- Un “alfa” es un elemento esencial de un esfuerzo de ingeniería de software, relevante para evaluar su progreso y salud
  - Oportunidad
  - Stakeholders (interesados)
  - Requisitos
  - Sistema de software
  - Trabajo
  - Forma de trabajo
  - Equipo

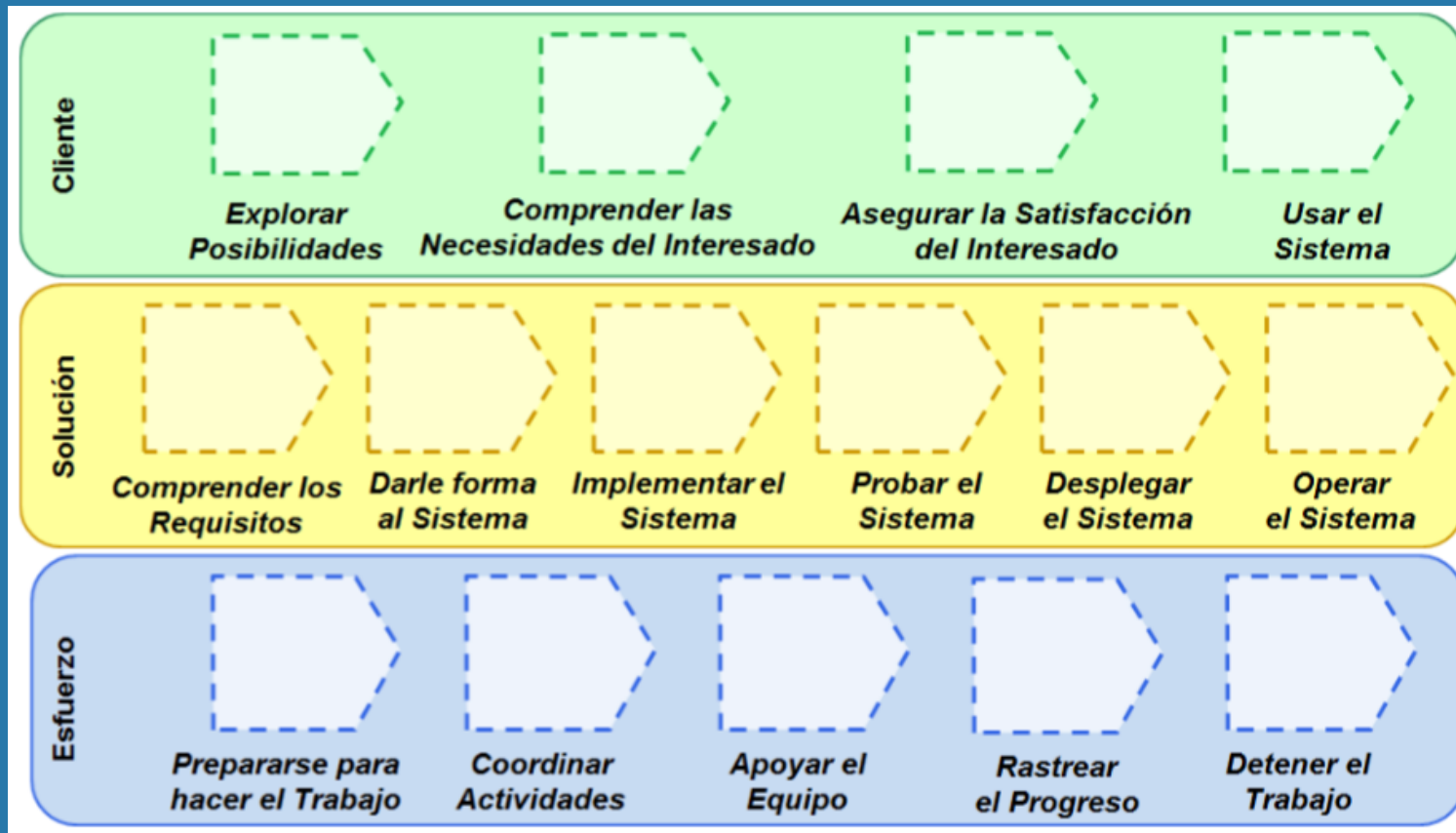
# Kernel: alfas [3]

- Los alfas tienen estados que representan su progreso y salud
  - sistema de software: “con arquitectura seleccionada”, “demostrable”, “usable”, “listo”, “operacional” y “retirado”
  - requisito: “concebido”, “acotado”, “coherente”, “aceptable”, “tratado”, “cumplido”
  - forma de trabajo: “con principios establecidos”, “con bases establecidas”, “en uso”, “en su lugar”, “trabajando bien”, “retirado”



# Kernel: prácticas [1]

- “Cosas que siempre hacemos”



# Kernel: prácticas [2]

- El Kernel es extensible y puede expandirse para apoyar diferentes proyectos
- El Kernel permite agregar *prácticas*, p.ej. historias de usuario, casos de uso, desarrollo basado en componentes, arquitectura, programación por pares, reuniones diarias de pie, equipos auto-organizados ...
- Es fundamental separar prácticas
  - Las prácticas se representan como unidades modulares separadas y distintas, que un equipo puede escoger si usa o no

# Kernel: tarjetas [1]



# Kernel: tarjetas [2]

- El Kernel es un marco de pensamiento *tangible y práctico* que apoya a los profesionales de software en su día a día
- Las tarjetas proporcionan recordatorios y señales concisos para los miembros del equipo en sus tareas diarias

# Aplicaciones del Kernel

- El núcleo apoyar la vida diaria de los profesionales del software
  - Hacer un sprint (o iteración)
  - Hacer un desarrollo completo, desde la idea hasta el producto
  - Escalar hacia grandes organizaciones y desarrollo de sistemas complejos

# Un ejemplo [1]

- [Jacobson et al., Communications of the ACM, 55(12), dec 2012, pp. 42-49]
- Una empresa tiene sus procesos formales muy poco desarrollados
- Tiene empleados con trayectoria. Los empleados nuevos, la mayoría recién titulados, tienen buenas habilidades técnicas (p.ej. en lenguajes de programación), pero no manejan tan bien otros aspectos del desarrollo, p.ej. trabajar con los stakeholders para lograr acuerdos en los requisitos

# Un ejemplo [2]

- La empresa tiene un equipo de desarrollo para hacer una aplicación móvil para redes sociales, que permita a sus usuarios compartir ideas, fotos y comentarios, y navegar por ellos
- El equipo comenzó con sólo dos desarrolladores: Smith (líder del equipo) y Tom; ambos conocían el Kernel
- Luego, se unieron otros dos desarrolladores (Dick y Harriet) nuevos en el trabajo y sin experiencia previa con el Kernel
- Para Smith, el éxito significaba más que funcionalidad, cronograma y calidad; este equipo hacía el desarrollo de forma iterativa

# Un ejemplo [3]

- Supongamos que:
  - Smith y su equipo llevan seis semanas en el desarrollo
  - Ya entregaron prototipos a los stakeholders
  - Los usuarios finales aún no pueden usar el sistema
- Hasta aquí, ¿cómo se puede utilizar el Kernel?



# Un ejemplo [4]

- Colocar las tarjetas de cada alfa en una fila de una tabla con el primer estado a la izquierda y el último estado a la derecha.
- Detenerse en cada estado y preguntar al equipo si ya se alcanzó; si es así, mover la tarjeta hacia la izquierda
- Continuar con la tarjeta siguiente hasta llegar al estado que el equipo aún no logra
- Mover las tarjetas de estado y el resto de las tarjetas de estados pendientes hacia la derecha
- Ver diapo siguiente: estados alcanzados por el equipo a la izquierda, y estados aún no alcanzados a la derecha (por simplicidad, se muestra sólo tres de los alfas)

# Un ejemplo [5]

<p><b>Requisitos</b></p> <p><b>Concebido</b></p> <ul style="list-style-type: none"> <li>La necesidad de un nuevo sistema es clara</li> <li>Se identificaron los usuarios</li> <li>Se identificaron los promotores iniciales</li> </ul> <p>1/6</p>	<p><b>Requisitos</b></p> <p><b>Acotado</b></p> <ul style="list-style-type: none"> <li>Se acordaron el propósito y la extensión del sistema</li> <li>Los criterios de éxito son claros</li> <li>Se acordaron los mecanismos para manejar los requisitos</li> <li>Se identificaron las restricciones y suposiciones</li> </ul> <p>2/6</p>	<p><b>Requisitos</b></p> <p><b>Coherente</b></p> <ul style="list-style-type: none"> <li>La visión general es clara y la comparten todos los involucrados</li> <li>Se explicaron importantes escenarios de uso</li> <li>Las prioridades son claras</li> <li>Se trataron los conflictos</li> <li>Se comprende el impacto</li> </ul> <p>3/6</p>	<p><b>Requisitos</b></p> <p><b>Aceptable</b></p> <ul style="list-style-type: none"> <li>Los requisitos describen una solución aceptable para los interesados</li> <li>La tasa de cambio para acordar requisitos es baja</li> <li>El valor es claro</li> </ul> <p>4/6</p>	<p><b>Requisitos</b></p> <p><b>Tratado</b></p> <ul style="list-style-type: none"> <li>Suficientes requisitos se implementaron para que el nuevo sistema sea aceptable</li> <li>Los interesados acuerdan que el sistema vale la pena realizando trabajo operativo</li> </ul> <p>5/6</p>	<p><b>Requisitos</b></p> <p><b>Cumplido</b></p> <ul style="list-style-type: none"> <li>El sistema satisface completamente los requisitos y las necesidades</li> <li>No hay ítems excepcionales de requisitos excepcionales que impidan que el sistema se considere completo</li> </ul> <p>6/6</p>
<p><b>Sistema de software</b></p> <p><b>Con arquitectura seleccionada</b></p> <ul style="list-style-type: none"> <li>Se seleccionó la arquitectura que trata los riesgos técnicos clave</li> <li>Se acordaron los criterios para seleccionar la arquitectura</li> <li>Se seleccionaron las plataformas, tecnologías y lenguajes</li> <li>Se tomaron las decisiones de compra, construcción y reuso</li> </ul> <p>1/6</p>	<p><b>Sistema de software</b></p> <p><b>Demostrable</b></p> <ul style="list-style-type: none"> <li>Se demostraron las características clave de la arquitectura</li> <li>Los interesados relevantes acordaron que la arquitectura es apropiada</li> <li>Se ejercieron la interfaz crítica y las configuraciones del sistema</li> </ul> <p>2/6</p>	<p><b>Sistema de software</b></p> <p><b>Usable</b></p> <ul style="list-style-type: none"> <li>El sistema es usable y tiene las características de calidad deseadas</li> <li>Los usuarios pueden operar el sistema</li> <li>Se aceptaron los niveles de defectos</li> <li>Se conoció el contenido de liberación</li> </ul> <p>3/6</p>	<p><b>Sistema de software</b></p> <p><b>Listo</b></p> <ul style="list-style-type: none"> <li>Se puso a disposición la documentación de usuario</li> <li>Los representantes de los interesados aceptaron el sistema</li> <li>Los representantes de los interesados quieren que se haga operacional el sistema</li> </ul> <p>4/6</p>	<p><b>Sistema de software</b></p> <p><b>Operacional</b></p> <ul style="list-style-type: none"> <li>El sistema se usó en un ambiente operacional</li> <li>El sistema está disponible para los usuarios previstos</li> <li>Al menos un ejemplo del sistema es completamente operacional</li> <li>El sistema es compatible con los niveles de servicio acordados</li> </ul> <p>5/6</p>	<p><b>Sistema de software</b></p> <p><b>Retirado</b></p> <ul style="list-style-type: none"> <li>No se da más soporte al sistema</li> <li>No se producirán más actualizaciones al sistema</li> <li>Se reemplazó o se discontinuó el sistema</li> </ul> <p>6/6</p>
<p><b>Forma de trabajo</b></p> <p><b>Con principios establecidos</b></p> <ul style="list-style-type: none"> <li>Se establecieron los principios y restricciones</li> <li>Se comprometieron los principios y restricciones</li> <li>Se acordaron las prácticas y herramientas</li> <li>Se comprendió el contexto en que el equipo debe operar</li> </ul> <p>1/6</p>	<p><b>Forma de trabajo</b></p> <p><b>Con bases establecidas</b></p> <ul style="list-style-type: none"> <li>Prácticas clave y herramientas listas</li> <li>Se analizaron y comprendieron las brechas entre prácticas y herramientas</li> <li>Se analizaron y comprendieron las brechas de capacidad</li> <li>Se integraron las prácticas seleccionadas y las herramientas</li> </ul> <p>2/6</p>	<p><b>Forma de trabajo</b></p> <p><b>En uso</b></p> <ul style="list-style-type: none"> <li>Algunos miembros del equipo están usando la forma de trabajo</li> <li>Se inspeccionó regularmente el uso de prácticas y herramientas</li> <li>El equipo está adaptando y apoyando las prácticas y herramientas</li> <li>Los procedimientos están en su lugar para manejar retroalimentación</li> </ul> <p>3/6</p>	<p><b>Forma de trabajo</b></p> <p><b>En su lugar</b></p> <ul style="list-style-type: none"> <li>Todos los miembros del equipo están usando la forma de trabajo</li> <li>Todos los miembros tienen acceso a las prácticas y herramientas para hacer su trabajo</li> <li>El equipo completo se involucró en la inspección y adaptación de la forma de trabajo</li> </ul> <p>4/6</p>	<p><b>Forma de trabajo</b></p> <p><b>Trabajando bien</b></p> <ul style="list-style-type: none"> <li>La forma de trabajo está trabajando bien para el equipo</li> <li>Los miembros del equipo están progresando según el plan</li> <li>El equipo aplica naturalmente prácticas sin pensar en ellas</li> <li>Las herramientas soportan naturalmente la forma de trabajo</li> </ul> <p>5/6</p>	<p><b>Forma de trabajo</b></p> <p><b>Retirado</b></p> <ul style="list-style-type: none"> <li>El equipo no usa más la forma de trabajo</li> <li>Se compartieron las lecciones aprendidas para uso futuro</li> </ul> <p>6/6</p>

# Un ejemplo [6]

- Pasos siguientes con el kernel
  - Determinar qué hacer
    - Discutir los próximos estados objetivos
    - Comprender los alfas seleccionados
  - Determinar cómo llegar a esos próximos estados objetivos
    - Establecer prioridades
    - Determinar forma de trabajo
    - Planear iteraciones

# Áreas de SEMAT

- Actualmente SEMAT se divide en 4 áreas ([www.semat.org](http://www.semat.org))
  - Área Práctica
  - Área de Educación
  - Área Teórica
  - Comunidad

# Áreas de SEMAT (Práctica)

- Desarrollo de un Kernel común que incluya la esencia de la Ingeniería de Software
- Establecer métodos específicos
- Crear lenguajes esenciales
- Publicación realizada en OMG (Object Management Group), Julio 2013

# Áreas de SEMAT (Educación)

- Aplicar SEMAT en la educación de la Ingeniería de Software
- Mezclar la industria con la academia
- Educación global
- Creación de materias y cursos globalmente intercambiables
- Primeros pilotos realizados en algunas universidades, tales como: Carnegie Mellon University Silicon Valley, Free University of Bozen Bolzano, KTH Royal Institute of Technology, Universidad Nacional de Colombia, otros

# Áreas de SEMAT (Teórica)

- Creación de una teoría general de Ingeniería de Software
- Teoría debe ser sólida para ingenieros y profesionales del área
- Predicción de problemas en nuestra área

# Áreas de SEMAT (Comunidad)

- Establecer entidades legales que utilicen SEMAT
- Creación de páginas web
- Creación de una comunidad global creciente de ayuda mutua



# Conclusiones

- SEMAT es una iniciativa que beneficia a la profesión
- El Kernel proporciona la base para la creación de cursos fundamentales de la ingeniería de software, que se pueden luego complementar con cursos adicionales sobre prácticas específicas (ya sea parte del currículum educativo inicial o después durante el desarrollo profesional de los estudiantes)



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA



**Departamento de Informática**  
Universidad Técnica Federico Santa María

# SEMAT

Ingeniería de Software – INF-225

**Hernán Astudillo & Gastón Márquez**

*Departamento de Informática*

*Universidad Técnica Federico Santa María*

<hernan@inf.utfsm.cl, g.marquez.o@gmail.com>