**Capítulo 2: Modelos Prescriptivos de Proceso de Desarrollo de Software**

**Introducción**

* Los modelos prescriptivos definen un conjunto distinto de actividades, acciones, tareas, fundamentos y productos de trabajo necesarios para desarrollar software de alta calidad.
* Incluyen actividades estructurales: comunicación, planeación, modelado, construcción y despliegue.

**Modelos de Proceso de Software**

1. **Modelo de Cascada (Waterfall Model)**
   * **Descripción**: Secuencial, cada fase debe completarse antes de pasar a la siguiente.
   * **Filosofía**: Enfoque ordenado y disciplinado.
   * **Etapas**: Requisitos, diseño, implementación, verificación, mantenimiento.
   * **Ventajas**: Simple y fácil de entender, útil para proyectos pequeños.
   * **Desventajas**: Poco flexible, difícil de retroceder a fases anteriores.
2. **Modelos de Proceso Incremental (Incremental Models)**
   * **Descripción**: Desarrollo en incrementos o versiones sucesivas.
   * **Filosofía**: Entrega parcial y continua de funcionalidades.
   * **Ventajas**: Flexibilidad, permite la retroalimentación temprana.
   * **Desventajas**: Requiere una buena planificación y diseño inicial.
3. **Modelos de Proceso Evolutivo (Evolutionary Process Models)**
   * **Prototipado**: Crear prototipos rápidos para refinar requisitos.
   * **Modelo Espiral**: Combina el desarrollo iterativo con la gestión de riesgos.
   * **Ventajas**: Adaptable a cambios, enfocado en la reducción de riesgos.
   * **Desventajas**: Puede ser costoso y complejo de gestionar.
4. **Modelos de Proceso Especializado**
   * **Modelo de Desarrollo Concurrente**: Actividades simultáneas en diferentes estados.
   * **Modelo de Proceso Formal**: Métodos matemáticos para especificar y verificar sistemas.
   * **Modelo Basado en Componentes**: Uso de componentes reutilizables.
5. **Proceso Unificado (Unified Process)**
   * **Descripción**: Iterativo e incremental, basado en la UML (Unified Modeling Language).
   * **Fases**: Inicio, elaboración, construcción, transición.
   * **Ventajas**: Amplia aceptación, bien documentado.
   * **Desventajas**: Puede ser pesado y complejo para proyectos pequeños.

**Capítulo 3: Ingeniería de Software Ágil**

**Introducción**

* La ingeniería de software ágil combina una filosofía y un conjunto de directrices de desarrollo.
* Busca la satisfacción del cliente mediante la entrega temprana y continua de software funcional.

**Principios de la Filosofía Ágil**

* **Satisfacción del cliente**: Entrega temprana y continua de software valioso.
* **Adaptación a los cambios**: Aceptar cambios de requisitos, incluso en etapas avanzadas.
* **Entregas frecuentes**: Preferencia por entregas cortas y frecuentes.
* **Colaboración**: Colaboración diaria entre desarrolladores y clientes.
* **Motivación**: Equipos de proyecto pequeños y motivados.
* **Comunicación**: Cara a cara es la forma más eficiente y efectiva.
* **Simplicidad**: Maximizar la cantidad de trabajo no hecho.
* **Autoorganización**: Los mejores diseños emergen de equipos autoorganizados.
* **Mejora continua**: Reflexionar y ajustar el comportamiento regularmente.

**Modelos de Procesos Ágiles**

1. **Programación Extrema (Extreme Programming, XP)**
   * **Principios**: Comunicación, simplicidad, retroalimentación, coraje, respeto.
   * **Prácticas**: Desarrollo dirigido por pruebas, programación en pareja, integración continua.
2. **Desarrollo Adaptativo de Software (Adaptive Software Development, ASD)**
   * **Fases**: Especulación, colaboración, aprendizaje.
   * **Enfoque**: Adaptabilidad y aprendizaje continuo.
3. **Método de Desarrollo de Sistemas Dinámicos (Dynamic Systems Development Method, DSDM)**
   * **Principios**: Entrega oportuna, participación del usuario, control del alcance.
   * **Fases**: Estudio de viabilidad, estudio de negocio, ciclo de desarrollo, implementación.
4. **Scrum**
   * **Roles**: Product Owner, Scrum Master, Development Team.
   * **Eventos**: Sprints, Daily Scrum, Sprint Review, Sprint Retrospective.
   * **Artefactos**: Product Backlog, Sprint Backlog, Incremento.
5. **Cristal (Crystal)**
   * **Enfoque**: Adaptabilidad a distintos tamaños y criticidad de proyectos.
   * **Variantes**: Crystal Clear, Crystal Yellow, Crystal Orange, etc.
6. **Desarrollo Impulsado por las Características (Feature Driven Development, FDD)**
   * **Fases**: Desarrollo global del modelo, construcción de una lista de características, planificación por características, diseño y construcción por características.
7. **Modelado Ágil (Agile Modeling, AM)**
   * **Prácticas**: Modelado justo a tiempo, modelos suficientemente buenos, equipo de desarrollo centrado.
8. **Desarrollo de Software Lean (Lean Software Development)**
   * **Principios**: Eliminar desperdicios, construir con calidad, crear conocimiento, diferir el compromiso, entregar rápido, respetar a las personas, optimizar el todo.