# Unidad 1: Ingeniería de Software en Contexto

## Software:

Escala de tiempo

Descripción generada automáticamente

* El Proyecto es el medio por el cual organizamos los recursos, asignar tareas y es el medio por el cual se entrega el producto de software al cliente.
* El Producto es el resultado del proyecto y los procesos, es lo que se entrega al cliente, en este caso, sería el software funcionando.
* Las Personas son el factor más importante en el software ya que determinan el éxito del proyecto de software.
* El Proceso es una descripción sobre lo que se debe hacer para construir el software. El proceso se adapta al proyecto.
* Las Herramientas nos ayudan a automatizar los procesos para que sean más eficientes y productivos.

Software es un conjunto de: programas, procedimientos, reglas, documentación y datos.

Software es conocimiento o información que se presenta en distintos niveles de abstracción.

La información tiene que estar estructurada con propiedades lógicas y funcionales. Además, es creada y mantenida en varias formas y representaciones. A su vez, es confeccionada para ser procesada por computadora en su estado más desarrollado.

## Ingeniería de software:

Es una disciplina que se encarga de todos los aspectos de la producción del software, desde la primera etapa de especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación.

## Disciplinas que conforman la ingeniería de software:

Diseño se enfoca en las disciplinas técnicas.

Disciplinas:

* Técnicas -> vinculada al producto de software. Nos referimos a actividades que aportan al desarrollo de software como producto.
  + Requerimientos.
  + Análisis y desarrollo. -> La materia diseño se centra en este apartado.
  + Implementación.
  + Prueba.
  + Despliegue.
* Gestión -> vinculada al proyecto. Hacen referencia a actividades de planificación, monitoreo y control del proyecto de desarrollo de software.
  + Planificación.
  + Monitoreo y control de proyecto (también le dicen, seguimiento y supervisión).
* Soporte o protectoras -> son disciplinas transversales.
  + Aseguramiento de calidad.
  + SCM: gestión de configuración de software.
  + Métricas.

## La crisis del software:

El software nunca está a la altura, no hay software de suficiente calidad para satisfacer las demandas que hay. Cada vez hay más demanda y más necesidad, por lo cual, siempre hay gente insatisfecha.

Muchos softwares que se hacen se tiran porque no cumplen los requerimientos. La mayoría de las veces buscamos soluciones técnicas a estos problemas cuando en realidad son problemas generados por los humanos en términos de comunicación, comprensión y manejo de expectativas.

Causas para “tirar” software:

* Introducción de nuevas tecnologías de hardware que generaron la posibilidad de construir software más complejo y de mayor tamaño.
* La complejidad que supone la tarea de programar.
* Los cambios a los que tiene que estar sometido el software para adaptarse a las necesidades del usuario.

## Cambios en el software:

Tienen su origen en:

* Cambios del negocio y nuevos requerimientos.
* Soporte de cambios de productos asociados.
* Reorganización de las prioridades de la empresa por crecimiento.
* Cambios en el presupuesto.
* Defectos encontrados a corregir.
* Oportunidades de mejora.

## Ciclo de vida:

Es la serie de pasos a través de los cuales el producto o proyecto progresa. Es decir, que es una representación simplificada de un proceso, el cuál define elementos del proceso y el flujo del proceso.

Los productos tienen su ciclo de vida, al igual que los proyectos.

Los ciclos de vida iterativos pueden ser de duración fija o de alcance fijo. El enfoque del ciclo de vida iterativo e incremental del proceso unificado es de alcance fijo, mientras que el enfoque del ciclo de vida iterativo e incremental de scrum es de alcance fijo.

En el proceso unificado el ciclo de vida termina cuando se logran los casos de uso, se plantea una duración inicial, pero si no se logra en ese tiempo la iteración se alarga.

En scrum o agile en general, se establece una duración fija y se plantean algunas características, sino se logra en ese tiempo desarrollar esas características se entrega con menos, pero la duración no se alarga, no se negocia. Esto se denomina timebox, el tiempo no se modifica.

### Relación: ciclo de vida del proyecto y del producto:

Diagrama

Descripción generada automáticamente

No existe un impacto entre los ciclos de vida de cada uno, sino que el ciclo de vida del producto siempre es mayor que el ciclo de vida del proyecto, ya que el ciclo de vida del proyecto dura lo que dura el desarrollo del software. En cambio, el ciclo de vida del producto dura hasta que el software se deje de utilizar.

Es posible que un producto tenga varios proyectos en su ciclo de vida, debido a, constantes cambios y/o actualizaciones que se vayan realizando. Por lo que, dentro del ciclo de vida del producto, se pueden desarrollar varios ciclos de vida de proyectos.

### Ciclos de vida de proyectos de software:

* Un ciclo de vida de un proyecto de software es una representación de un proceso. Grafica una descripción del proceso desde una perspectiva particular.
* Los modelos especifican:
  + Las fases de proceso.
  + El orden en el cual se llevan a cabo.

### Clasificación de los ciclos de vida:

* Hay 3 tipos básicos de ciclos de vida para un desarrollo de software:
  + Secuencial.
  + Iterativo.
  + Recursivo.

Diagrama

Descripción generada automáticamente

#### Ciclo de vida secuencial:

Este modelo dispone de las actividades de forma lineal, es decir, que el proyecto progresa a través de una secuencia ordenada de pasos (fases) y una actividad no puede iniciar sin que la precedente haya sido finalizada.

#### Ciclo de vida iterativo/incremental:

Este modelo aplica sucesivas iteraciones en forma escalonada a medida que avanza el calendario de actividades. Cada iteración produce un incremento de software funcional potencialmente entregable. Usualmente los primeros incrementos incluyen las funciones básicas/críticas que más requiere el cliente.

El sistema se desarrolla como una serie de versiones (incrementos) y cada una añade funcionalidades a la versión anterior.

#### Ciclo de vida recursivo:

El modelo recursivo es utilizado para gestionar los riesgos del desarrollo de sistemas complejos a gran escala.

Para llevarlo adelante, se subdivide el proyecto en varios mini proyectos, intentando resolver en cada uno los riesgos más relevantes hasta que no quede ninguno.

Dicho en otras palabras, se inicia con algo en forma completa, como una subrutina que se llama a sí misma e inicia nuevamente. Se presenta un prototipo que va mejorando con cada vuelta. Se generan productos independientes de la implementación, que pueden ser reusables en sistemas de características similares.

## Proyecto:

Un proyecto es una unidad de gestión, que toma recursos y el tiempo con el objetivo de obtener un producto o un servicio.

**Características:**

* El resultado debe ser único.
* Tiene una fecha de inicio y una fecha de fin identificadas.
* Elaboración gradual: el proceso es el que hace la descomposición del proyecto.
* Se trabaja con tareas interrelacionadas.
* **Orientación a objetivos:**
  + Los proyectos están dirigidos a obtener resultados y ellos refleja a través de objetivos.
  + Los objetivos guían al proyecto.
  + Los objetivos no deben ser ambiguos.
  + Un objetivo claro no alcanza, debe ser también alcanzable.
* **Duración limitada:**
  + Los proyectos son temporarios, cuando se alcanza el/los objetivo/s, el proyecto termina.
  + Una línea de producción no es un proyecto.
* **Tareas interrelacionadas basadas en esfuerzos y recursos:**
  + Complejidad sistémica de los problemas.
* **Son únicos:**
  + Todos los proyectos por similares que sean tienen características que los hacen únicos.

### Administración de proyectos:

* Administración de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para satisfacer los requerimientos del proyecto.
* Administrar un proyecto incluye:
  + Identificar los requerimientos.
  + Establecer objetivos claros y alcanzables.
  + Adaptar las especificaciones, planes y el enfoque a los diferentes intereses de los involucrados (stakeholders).

### La restricción triple:

* Objetivos de proyecto (alcance): ¿qué está el proyecto tratando de alcanzar?
  + Son los requerimientos del proyecto, es decir, los límites o el ámbito sobre el cual se va a desarrollar el mismo.
* Tiempo ¿cuánto tiempo debería llevar completarlo?
  + Es la especificación de las fechas para realizar las entregas que determinarán el avance del proyecto.
* Costos: ¿cuánto debería costar?
  + Son los recursos implicados en el desarrollo del proyecto.
* El balance de estos tres factores afecta directamente la calidad del proyecto.
  + Proyectos de alta calidad entregan el producto requerido, el servicio o resultado, satisfaciendo los objetivos en el tiempo estipulado y con el presupuesto planificado.
* Es responsabilidad del líder de proyecto balacear estos tres objetivos (que a menudo compiten entre ellos).

Diagrama

Descripción generada automáticamente

Lo ideal para tener una ganancia económica es que se tengamos la posibilidad de modificar 2 de las 3, lo más sano es que el tiempo y el costo lo manejemos nosotros, los desarrolladores, y el cliente maneje los alcances.

En los procesos definidos parte desde que los alcances ya estén plenamente definidos, pudiendo modificar el costo y el tiempo.

En los procesos agiles parte desde que hay una IDEA de lo que se quiere, se parte desde un equipo ya definido (sería los recursos fijos) y el tiempo es fijo (el scrint es fijo). Luego de tener los recursos y el tiempo fijos, se ponen de acuerdo (desarrolladores y cliente) para establecer los que se va a desarrollar para esa iteración (lo que se va a desarrollar se saca del producto backlog).

El problema que tienen los procesos agiles es planear el contrato con el cliente, porque no está definido ni el producto ni el tiempo para el momento en donde se hace el contrato inicial. Los clientes no están acostumbrados a esta clase de contratos, ya que es un producto incremental.

### Rol del líder de proyecto/equipo:

Diagrama

Descripción generada automáticamente

Encargado de definir el alcance del proyecto, detallar las tareas estimando tiempo y requerimientos, identificar recursos y definir presupuestos, identificar y evaluar riesgos, preparar planes de contingencia, participar en las revisiones de las fases del proyectos.

### ¿Qué es un equipo de proyecto?

Un grupo de personas comprometidas en alcanzar un conjunto de objetivos de los cuales se sienten mutuamente responsables.

#### Características de un equipo de proyecto:

* Diversos conocimientos y habilidades.
* Posibilidad de trabajar juntos efectivamente/desarrollar sinergia.
* Usualmente es un grupo pequeño.
* Tienen sentido de responsabilidad como una unidad.

### Stakeholders:

Son los interesados del proyecto, incluye el equipo de proyecto, el equipo de dirección, el líder de proyecto y el patrocinador.

### Plan de proyecto:

* Un plan es a un proyecto lo que una hoja de ruta a un viaje.
* El plan de proyecto documenta:
  + ¿qué es lo que hacemos?
  + ¿cuándo lo hacemos?
  + ¿cómo lo hacemos?
  + ¿quién lo va a hacer?

Objetivos del proyecto deben ser claros y alcanzables.

El plan de proyecto consta de:

1. Objetivo del proyecto.
2. Alcance del proyecto.
3. Equipo de proyecto.
4. Proceso y ciclo de vida.
5. Estimaciones.
6. Riesgos.
7. Recursos.
8. Programación/calendarización.
9. Métricas.
10. Informes y reuniones.

Estas 2 ultimas (9 y 10) es la planificación del seguimiento.

#### ¿Qué implica la planificación de proyectos de software?

* Definición del alcance del proyecto.
* Definición de proceso y ciclo de vida.
* Estimación.
* Gestión de riesgos.
* Asignación de recursos.
* Programación de proyectos.
* Definición de controles.
* Definición de métricas.

#### Definición del alcance:

* **Alcance del producto:**
  + Son todas las características que pueden incluirse en un producto o servicio.
* **Alcance del proyecto:**
  + Es todo el trabajo y solo el trabajo que debe hacerse para entregar el producto o servicio con todas las características y funciones especificadas.

Todo el trabajo y solo el trabajo que tengo que hacer para cumplir con el objetivo.

##### ¿Cómo se mide el alcance?

* El cumplimiento del alcance del proyecto:
  + Se mide contra el plan de proyecto (o plan de desarrollo de software).
* El cumplimiento del alcance del producto:
  + Se mide contra la especificación de requerimientos.

### Definir un ciclo de vida:

Diagrama

Descripción generada automáticamente

## Procesos definidos vs procesos empíricos:

Existen dos tipos de procesos:

* Procesos definidos:
  + Un proceso definido asume que podemos repetir el mismo proceso una y otra vez, indefinidamente, y obtener los mismos resultados.
  + Hay un contexto organización, que define la forma de trabajar, y espera que los proyectos que se vayan a llevar adelante sigan con esta forma de trabajar.
  + Hay una secuencia definida de pasos, se van adaptando las necesidades de cada proyecto en particular.
  + Tiene una retroalimentación más tardía.
  + Lo que buscan es poder repetibilidad, es poder estimar nuevos proyectos a partir de proyectos pasados. Ejemplos, PDU/RUP.
* Procesos empíricos:
  + Se basan en la experiencia, pero plantean una forma de generar esa experiencia.
  + Sostiene que la experiencia no es extrapolable, es decir, la experiencia de uno no se transmite a los demás.
  + Plantean que la mejor forma de tomar decisiones es que la realice el que va a hacer el proyecto. El equipo define como cree que es la mejor forma de construir software.
  + Los equipos tienen que funcionar autónomamente. Aprender de la experiencia, la experiencia obtenida por el equipo.
  + El empirismo trabaja con 3 pilares que son los que determinan que esto pueda funcionar, estos pilares son los que toman el agilismo:
    - Inspección: paro lo que se está haciendo y reviso. Se inspecciona el proceso (se hace con el equipo) y el producto (se hace con el cliente).
    - Adaptación: adaptarse a los cambios detectados en la inspección.
    - Transparencia: la información debe ser visible y disponible para todas las personas implicadas en el proceso.

En la inspección y adaptación es donde se da la iteración, la realimentación es la experiencia.

El proceso se adapta al proyecto.