## El proceso de software:

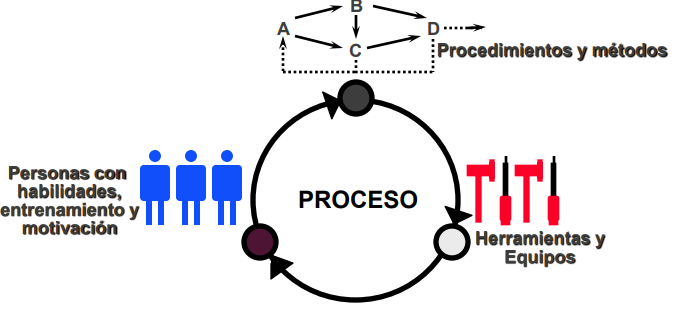
Diagrama

Descripción generada automáticamente

* Conjunto estructurado de actividades para desarrollar un sistema de software.
* Estas actividades varían dependiendo de la organización y el tipo de sistema que debe desarrollarse.
* Debe ser explícitamente modelado si va a ser administrado.

### Definición de un proceso de software:

* Proceso: es la secuencia de pasos ejecutados para un propósito dado.
* Proceso de software: es un conjunto de actividades, métodos, prácticas, y transformaciones que la gente usa para desarrollar o mantener software y sus productos asociados.



Se tienen tres factores determinantes del proceso:

* Procedimientos y métodos: se deben tener los procedimientos y métodos a utilizar definidos y escritos.
* Personas motivadas, capacitadas y con habilidades: las personas deben estar capacitadas y con habilidades para realizar sus tareas asignadas de la manera correcta, y deben estar motivadas para lograr una mayor eficiente.
* Herramientas y equipos: materiales necesarios para llevar a cabo el proceso que nos permiten que las entradas se transformen en salidas.

### Proceso definido:

Estos procesos consideran que ante la misma entrada se pretende que se obtendrá la misma salida.

* Asume que podemos repetir el mismo proceso una y otra vez, indefinidamente, y obtener los mismos resultados.
* La administración y control provienen de la predictibilidad del proceso definido.

### Proceso empírico:

* Asume procesos complicados con variables cambiantes. Cuando se repite el proceso, se pueden llegar a obtener resultados diferentes.
* La administración y control se realiza a través de inspecciones frecuentes y adaptaciones.
* Son procesos que trabajan bien con procesos creativos y complejos.

### 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.

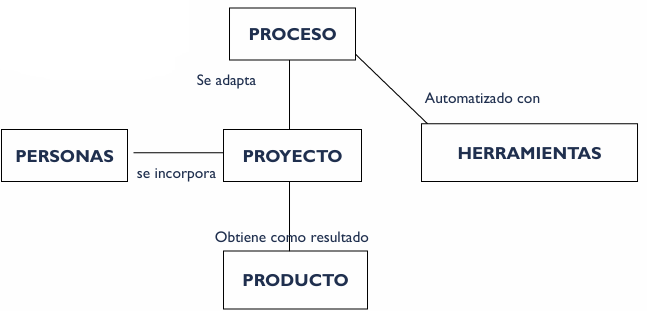
Es posible que un producto tenga varios proyectos en su ciclo de vida, debido a, constantes cambios y/o actualizaciones que se vayan realizando. Por lo que, dentro del ciclo de vida del producto, se pueden desarrollar varios ciclos de vida de proyectos.

#### 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.



##### 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?
* Tiempo ¿cuánto tiempo debería llevar completarlo?
* Costos: ¿cuánto debería costar?
* 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 lo ideal 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

### ¿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.

### 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

### Estimaciones de software:

Las estimaciones son difíciles porque derivan de los requerimientos.

Hay un conjunto de cosas que hay que estimar y en un orden dado:

1. Tamaño. ¿Qué tamaño tiene eso que quiero construir?. En el desarrollo tradicional podemos medir el tamaño con líneas de código (fue lo primero que se eligió para estimar tamaño, no sirve), alcances, requerimientos, casos de uso. Lo que se está usando en estos últimos tiempos son los casos de uso por complejidad para el desarrollo tradicional.
2. Esfuerzo. Se mide en horas/personas lineales (asumo cuantas horas necesito para construir eso en base al tamaño teniendo en cuenta que estoy solo). El esfuerzo se deriva del tamaño.
3. Calendario/tiempo. Se tiene en cuenta los días y cantidad de horas que se trabaja semanalmente, la cantidad de personas, si se pueden realizar tareas simultaneas.
4. Costo. Es una variable económica. La valorización se hace al final. Es el costo del esfuerzo, es el costo más representativo del proyecto del software, es el 80% del costo total del software.
5. Recursos críticos.

Son las estimaciones en el contexto del plan de proyecto.

### Riesgo:

El riesgo es probabilidad de la ocurrencia de una perdida o daño que impacte en el cumplimiento del proyecto. Siempre el riesgo tiene implicada probabilidad, si hay certeza es un problema no un riesgo.

El riesgo se mide por 2 variables o dimensiones:

* Probabilidad.
* Impacto: es la que mide cuanto daño puede ocasionar si ocurre.

Probabilidad x impacto = exposición.

La exposición es la forma que tenemos de cuantificar los riesgos, comparar y decidir de cuales nos vamos a ocupar.

Los proyectos más importantes son:

* Que cambien los requerimientos.
* Que se vaya alguien vital para el proyecto antes que termine el mismo.
* Riesgos que tengan que ver con la tecnología, probabilidad que hoy ante y que mañana no.

#### Gestión de riesgos:

Diagrama

Descripción generada automáticamente

Identificar y gestionar riesgos a través de todas las fases del proyecto.

Gestión de riesgos reactiva: espera a que el riesgo se transforme en problemas, y recién ahí actuar.

Gestión de riesgos proactiva: se anticipa a los riesgos, actúa antes que ocurran realizando un plan de mitigación.

### Recursos:

Los recursos críticos son los que normalmente es escaso, hay más demanda que oferta.

## Métricas de software:

* El dominio de las métricas del software se divide en:
  + Métricas de proceso.
  + Métricas de proyecto.
  + Métricas de producto.
* Las métricas del proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software.

Diagrama

Descripción generada automáticamente

### Métricas básicas para un proyecto de software:

* Tamaño del producto.
* Esfuerzo.
* Tiempo (calendario).
* Defectos.

#### Métricas por rol?

**Desarrollador:**

1. Esfuerzo.
2. Esfuerzo y duración estimada y actual de una tarea.
3. % de cobertura por el unit test.
4. Número y tipo de defectos encontrados en el unit test.
5. Número y tipo de defectos encontrados en revisión por pares.

**Organización:**

1. Tiempo calendario.
2. Performance actual y planificada de esfuerzo.
3. Performance actual y planificada de presupuesto.
4. Precisión de estimaciones en Schedule y esfuerzo.
5. Defectos en release.

**Equipo de desarrollo:**

1. Tamaño del producto.
2. Duración estimada y actual entre los hitos más importantes.
3. Niveles de staffing actual y estimados.
4. Número de tareas planificadas y completadas.
5. Distribución del esfuerzo.
6. Status de requerimientos.
7. Volatibilidad de requerimientos.
8. Número de defectos encontrados en la integración y prueba de sistemas.
9. Número de defectos encontrados en peer reviews.
10. Satus de distribución de defectos.
11. % de test ejecutados.

### Tres factores para el éxito de un proyecto:

* Monitoreo y feedback.
* Tener una misión/objetivo claro.
* Comunicación.

### Causas de fracasos en proyectos:

* Fallas al definir el problema.
* Planificar basado en datos insuficientes.
* La planificación la hizo el grupo de planificaciones.
* No hay seguimiento del plan de proyecto.
* Plan de proyecto pobre en detalles.
* Planificación de recursos inadecuada.
* Las estimaciones se basaron en “supuestos” sin consultar datos históricos.
* Nadie estaba a cargo.