

Curso de Ingeniería de Software

Unidad 4

Administración del proyecto

Guadalupe Ibargüengoitia

Hanna Oktaba

Objetivos

- Enseñar **conceptos y prácticas** relacionadas con la administración de un proyecto:
 - **Planificación** del proyecto
 - **Definición y planificación** de iteraciones
 - **Ejecución, seguimiento y control** de actividades

Entradas a esta unidad

- **Condiciones**
 - Equipo conformado
- **Productos de trabajo**
 - *Planteamiento de Necesidades*
 - Medios de comunicación inicializados
 - Herramienta(s) colaborativa(s) inicializada (s)
 - *Acuerdos del equipo*

DEFINICIONES DE CONCEPTOS

Concepto de Proyecto

- Un *proyecto* se puede definir como el esfuerzo de una organización para desarrollar, introducir o evaluar un resultado único.
- Es un esfuerzo que se lleva a cabo durante un periodo de tiempo delimitado, con un equipo de trabajo asignado y con recursos disponibles para producirlo. (PMBOK, 2013)

Administración de Proyecto

- La **Administración de Proyecto** es la aplicación de **conocimientos, habilidades, herramientas y técnicas** a las actividades de un proyecto para **satisfacer sus requisitos**. (PMBOK, 2013)
- **Requisitos** principales de un proyecto son: **calidad, tiempo y costo**.



Administración del proyecto de software

- Tiene por objetivo establecer y llevar a cabo sistemáticamente un conjunto de actividades a fin de lograr el objetivo del proyecto de software en calidad, tiempo y costo.

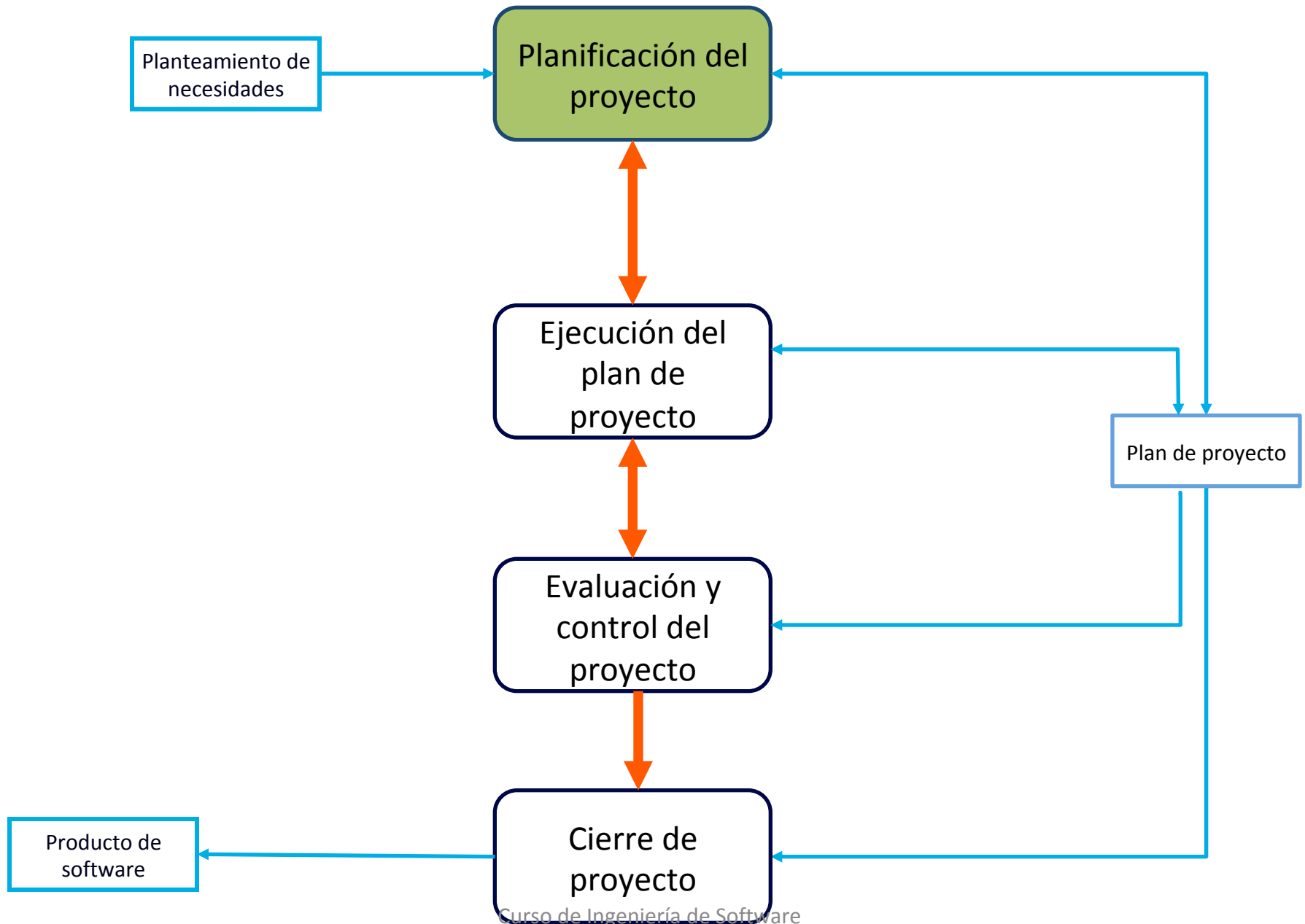
(ISO/IEC 29110)

Administración del proyecto

ISO/IEC 29110 Perfil básico

- Entrada: *Planteamiento de Necesidades*
- Actividades principales son:
 - Planificar el proyecto
 - Ejecutar el plan del proyecto
 - Evaluar y controlar el proyecto
 - Cerrar el proyecto
- Salida: Producto de Software documentado y funcional

Actividades del proceso de administración del proyecto



Planificar el proyecto

- ¿Qué debe de saber el equipo para iniciar un proyecto?

Plan del proyecto

- Cliente
- Objetivos del proyecto basados en las necesidades del cliente
- Requerimientos funcionales básicos del software
- Requerimientos no funcionales del software
- Lista priorizada de requerimientos de proyecto (LPRP)
- Restricciones (legales, técnicas, de seguridad,...).
- Fechas de inicio y fin del proyecto
- Forma de trabajar acordada
- El número de iteraciones y su duración
- Lista de los entregables – documentación y código de software
- Fechas y forma de entrega
- Integrantes del proyecto y sus roles

¿Cómo definir los requerimientos funcionales básicos del software?

- El **diagrama general de casos de uso** es una de las técnicas para identificar los requerimientos funcionales de un producto de software (Jacobson I., 2012)
- Es una vista gráfica más general y abstracta del sistema de software.
- Representa las funcionalidades básicas del software.
- Define qué tipo de usuarios utilizarán el software.

Diagrama general de casos de uso

- Esta técnica se utiliza para comprender las necesidades o los requerimientos funcionales y a partir de ellos se diseña, implementa y prueba el software.
- La importancia del *Diagrama general de casos de uso* reside en que las prácticas de desarrollo se pueden guiar por los casos de uso (Rosenberg Doug, 2001)

¿Qué es un caso de uso?

- Es la **representación** de la **manera** en la cual un **usuario** particular va a **lograr** un **objetivo** al **usar** un sistema de **software**.

Componentes gráficos de un Caso de uso

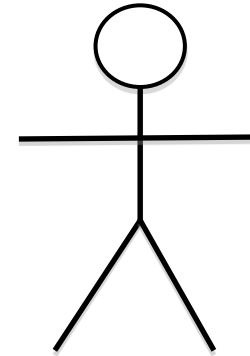
- Caso de uso

- Funcionalidad

- Actor

- Ejecutante de la(s) funcionalidad(es)

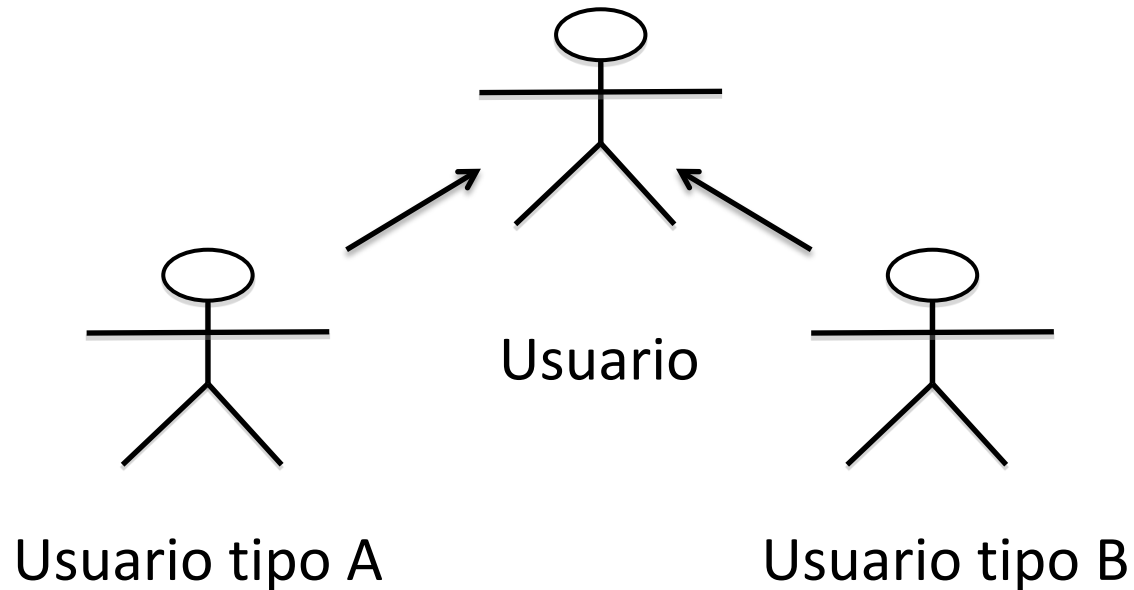
- Relación de asociación



Usuario

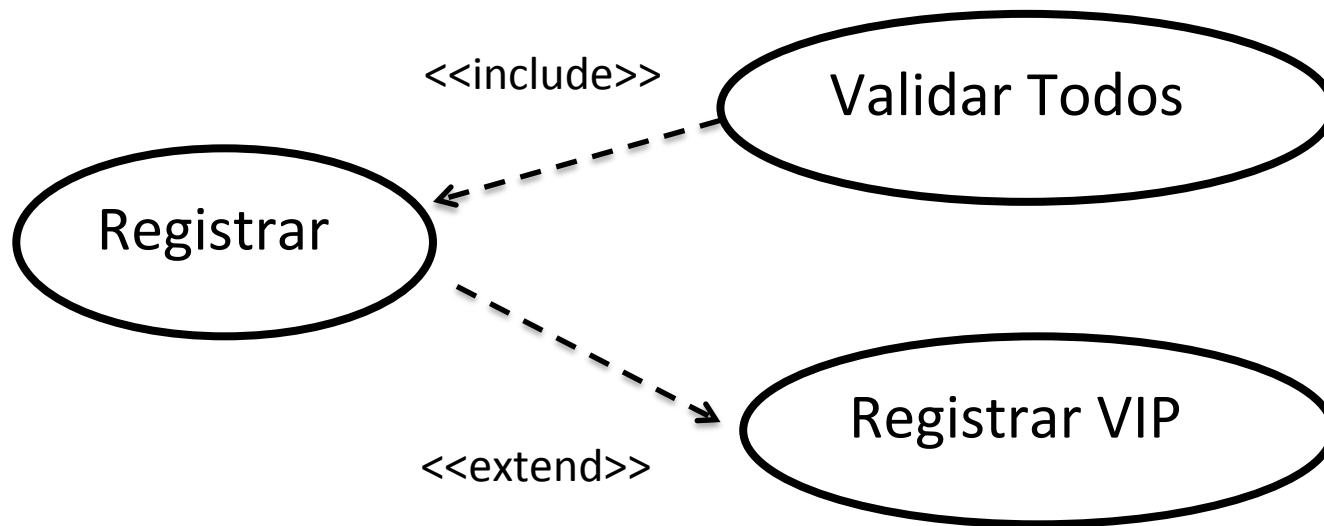
Componentes de gráficos un Caso de uso

- Relación de herencia etre Actores



Componentes gráficos de un Caso de uso

- Relaciones de inclusión o extensión entre casos de uso

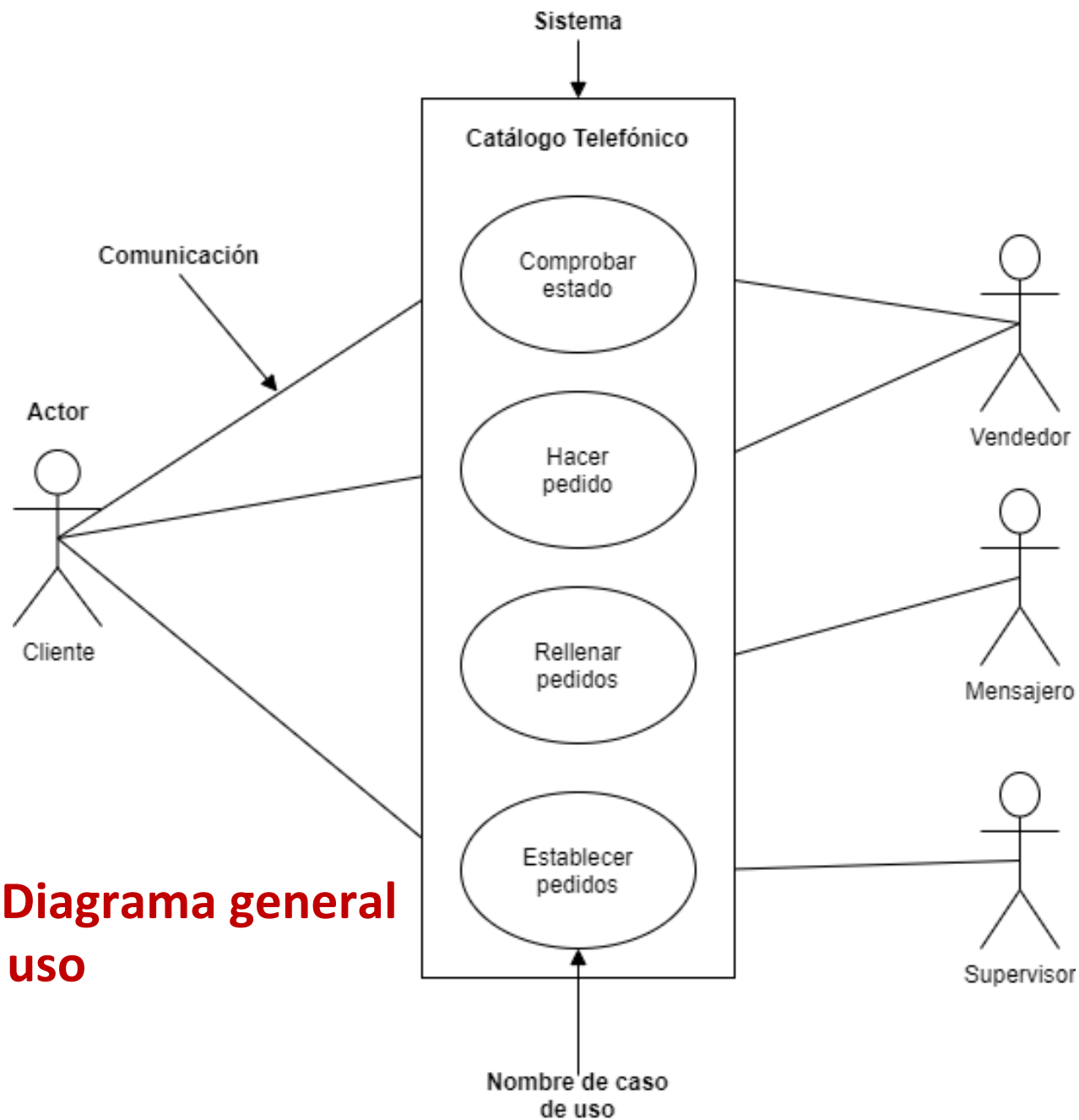


Convención de nombres

- **Actor – sustantivo** genérico, inicia con **mayúscula**, ejemplos: Alumno, Comprador, Visitante...
- **Caso de uso – empieza con verbo en infinitivo**, inicia con **mayúscula**, máximo dos-tres palabras, ejemplos: Registrar usuario, Consultar catálogo, Modificar perfil,...

Patrón de casos de uso CRUD

- Es un caso de uso para administrar la información
- Su nombre proviene de:
Create, Read, Update, Delete
Alta, Consulta, Cambio/Edición, Baja



Ejemplo del Diagrama general de casos de uso

¿Qué son los requerimientos no funcionales?

- Describen las características deseadas de software que no están relacionadas directamente con la funcionalidad.
- El estándar ISO/IEC 25010 conocido como SQuaRe (Software Quality ReQuirements) proporciona un modelo de las características de calidad de software. La primera cualidad habla de la funcionalidad pero las demás hablan de las cualidades no funcionales.

Cualidades del software (ISO/IEC 25010)

- **Adecuación Funcional** (*Functional suitability*). Representa la capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.
- **Eficiencia de Desempeño** (*Performance efficiency*). Representa el desempeño relativo a la cantidad de recursos utilizados, bajo determinadas condiciones.
- **Compatibilidad** (*Compatibility*). Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno de hardware o software.
- **Usabilidad** (*Usability*). Capacidad del producto de software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Cualidades del software (ISO/IEC 25010)

- **Fiabilidad** (*Reliability*). Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.
- **Seguridad** (Security). Capacidad de protección de la información y los datos de manera, que personas o sistemas no autorizados no puedan leerlos o modificarlos.
- **Mantenibilidad** (Maintainability). Capacidad del producto de software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
- **Portabilidad** (Portability): Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno operacional o de uso, de hardware y software a otro.

¿Porqué necesitamos la lista priorizada de requerimientos??

- El que **prioriza** los requerimientos es el **Cliente**
- El Cliente determina el **orden de los requerimientos** por el **valor que aportan** a su organización
- Representa la **visión y expectativas** del Cliente respecto a los **objetivos y entregas** del producto o proyecto

¿Qué es el proceso de ciclo de vida de software?

- El **proceso de ciclo de vida del software** es un conjunto de métodos, prácticas, actividades y transformaciones, que utilizan las personas para **desarrollar y mantener software** y sus **productos asociados**. (SWEBOOK)
- Un proyecto abarca una **porción del ciclo de vida de software** y, por lo general, está enfocado **al desarrollo nuevo** o al **mantenimiento del software** ya existente.

¿Qué significa acordar la forma de trabajar y porqué es importante ?

- Acordar la forma de trabajar para un proyecto es definir qué métodos, prácticas y actividades van a seguir las personas para desarrollarlo.
- Es importante acordarlo para que todos miembros del equipo sigan los mismos lineamientos de trabajo y se eviten conflictos y problemas de calidad en los resultados.

¿Qué es una iteración?

- El **proceso de ciclo de vida** de software **adaptativo** se caracteriza por el **desarrollo iterativo**, es decir se generan productos de software en ciclos de desarrollo o mantenimiento **frecuentes**, llamadas **iteraciones**. (SWEBOK)
- En **SCRUM** las **iteraciones** se llaman **Sprints**.
- Al **final de cada iteración** se debe **entregar** un **software ejecutable** para que sea **revisado** por el **cliente**.
- Las **sugerencias de cambios** y **nuevas funcionalidades** se agregan en **iteraciones posteriores**.

Iteraciones y el Plan del proyecto

- En el Plan del proyecto se define la duración y el número de iteraciones en función tiempo total disponible y las necesidades del Cliente.
- La duración recomendada es de una a cuatro semanas.
- En cada iteración se realizan las actividades que permiten entregar un producto con funcionalidades parciales, que el Cliente puede probar.
- En cada iteración se incrementan y/o refinan las funcionalidades del software hasta satisfacer las necesidades del cliente.

¿Qué es un entregable?

- **Entregable** – es un producto de software único y verificable para realizar un servicio, que está sujeto a la aprobación del patrocinador del proyecto o del cliente (SEVOCAB)
- **Entregables** son los productos de trabajo de las actividades del proyecto (por ejemplo, documentos de diseño de arquitectura de software, informes de inspección, software probado). Se deben identificar y caracterizar. (SWEBOOK)

¿Qué hemos aprendido?

- ¿Para qué sirve planificar el proyecto?
- ¿Qué elementos se identifican en el plan del proyecto?

Planificar la iteración

- El **equipo** y el **Cliente** seleccionan, de común acuerdo, los requerimientos de la **Lista priorizada (LPRP)** definiendo de esta manera el alcance de la iteración.
- El **compromiso** del equipo es **implementar** estas funcionalidades **durante la iteración**.
- Para **cumplir el compromiso** se programan las **actividades** utilizando el **tablero Kanban**.

¿Qué es un tablero Kanban?

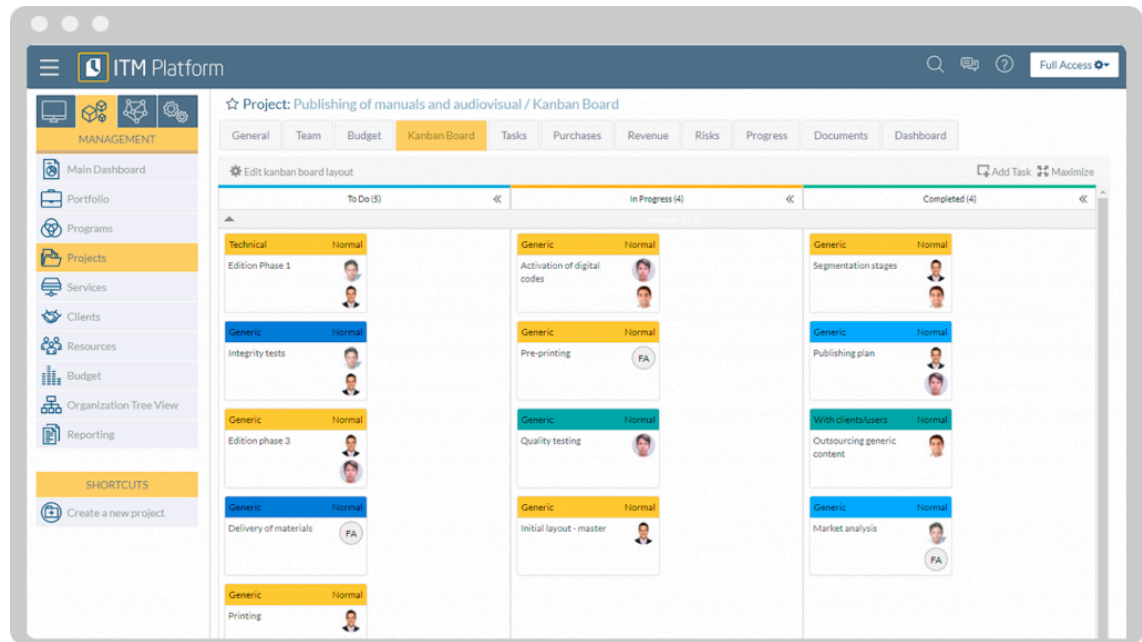
- El tablero **Kanban** es una técnica para **visualizar** el **flujo de trabajo** de cada miembro del equipo, creada en **Toyota**.
- **Kan**- visual, **ban**-tarjeta
- El objetivo del tablero Kanban es **planificar y dar seguimiento al avance** de las actividades de todos los miembros del equipo y poder ver **quién** está trabajando en qué parte del proyecto.

¿Qué es un tablero Kanban?

- El tablero (simplificado) consta de tres columnas con los encabezados:
 - *Por hacer* (To Do)
 - *Haciendo* (Doing)
 - *Hecho* (Done)
- En la columna de *Por hacer* se van colocando tarjetas que representan *cosas que se deben hacer* en la iteración.
- Cuando alguien empieza a trabajar mueve la tarjeta a la columna *Haciendo*.
- Cuando termina, la mueve a *Hecho*.

Tarjeta de trabajo

- Una tarjeta tiene:
 - Descripción de **qué es lo que hay que hacer**
 - **Responsable** de realizar la actividad
 - **Fecha de entrega**, para cuando se espera que el trabajo esté terminado.



Planificar la iteración

- Para planificar la iteración se colocan en la columna *Por hacer* del tablero Kanban las tarjetas que representan los requerimientos y otros compromisos, que deben realizarse durante la iteración.
- Un tablero puede tener diferentes números de columnas y renglones en función de la necesidad del proyecto.

Tablero Kanban del curso

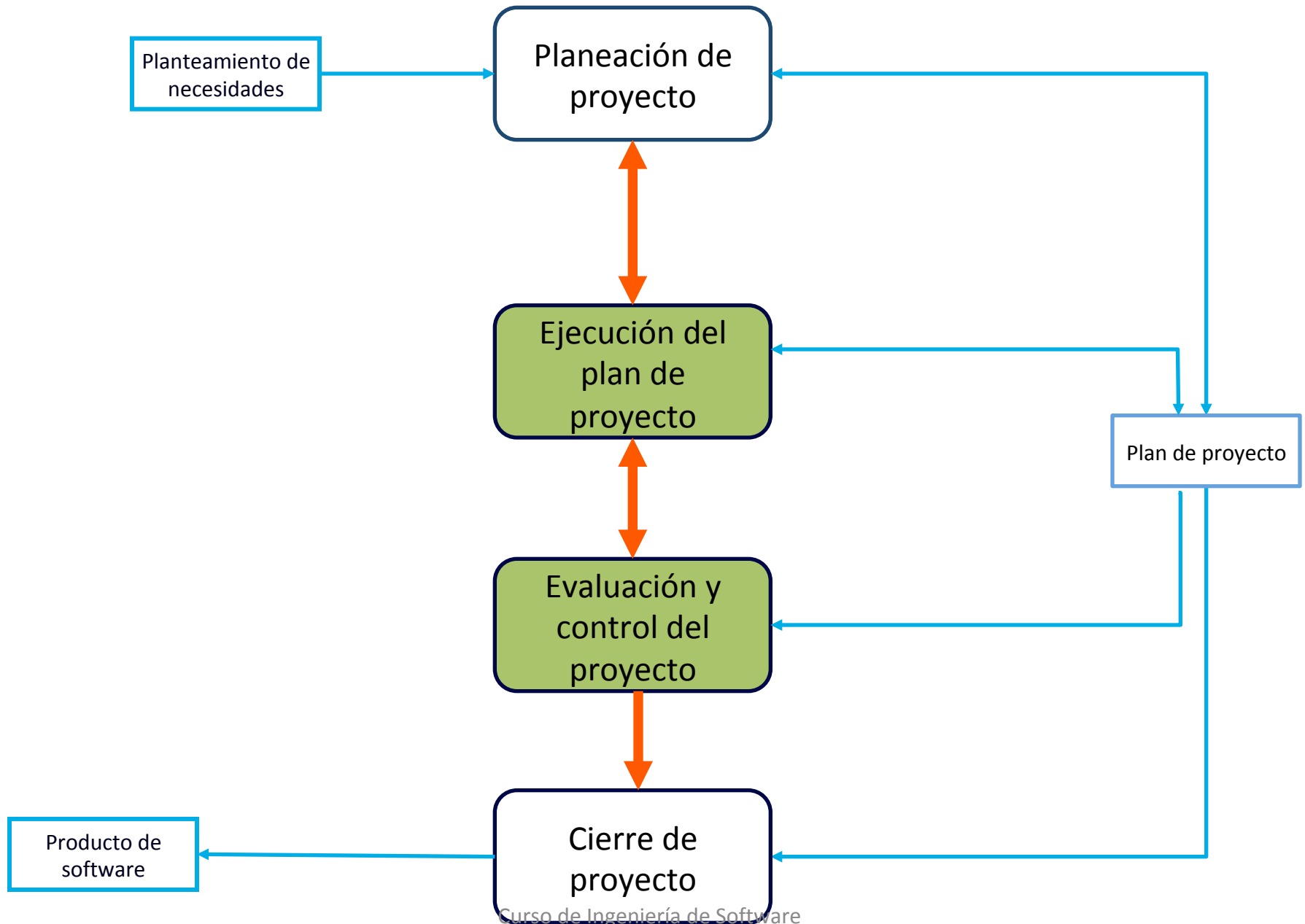
This screenshot shows the GitLab interface for a project named 'espacio-equip-1'. The top navigation bar includes the GitLab logo, project navigation links (Projects, Groups, Activity, Milestones, Snippets), a search bar, and user profile icons. The left sidebar shows the project structure with 'Issues' selected and a count of 1. The main area displays the 'Issue Boards' for 'Administrador > espacio-equip-1'. A search bar and 'Add list'/'Add issues' buttons are at the top. The Kanban board has three columns: 'To Do' (orange header), 'Doing' (green header), and 'RQA' (blue header). Each column has a trash icon, a counter showing '0', and a plus icon for adding new issues.

This screenshot shows the same GitLab interface for 'espacio-equip-1'. The Kanban board now includes two additional columns: 'RCLI' (red header) and 'Closed' (grey header). The 'To Do' and 'Doing' columns from the previous view are still present. All columns maintain the same layout with a trash icon, a counter showing '0', and a plus icon.

¿Qué hemos aprendido?

- ¿Qué se tiene que hacer para planificar una iteración?
- ¿Qué es un tablero Kanban y de qué sirve?

Actividades del proceso de administración del proyecto



Ejecutar el plan, dar seguimiento y controlar la iteración

- Los miembros del equipo reparten los compromisos de la iteración de manera equitativa, aprovechando conocimientos y experiencia de cada uno.
- Empiezan a trabajar de manera individual.
- Durante la reunión diaria se da seguimiento al cumplimiento de los compromisos.
- En el caso de que se identifiquen problemas, el equipo trata de controlarlos apoyándose mutuamente o buscando apoyo de los docentes.
- El avance del trabajo y las nuevas tareas se representan actualizando el tablero Kanban.

¿Qué hemos aprendido?

- ¿Para qué se tiene que dar el seguimiento y controlar el cumplimiento de los compromisos de los miembros del equipo?

PRACTICA ADMINISTRATIVA PARA PLANIFICAR EL PROYECTO

PA1 Planificar el proyecto

Objetivos

- Generar el **plan del proyecto** para guiar su realización.

PA1 Planificar el proyecto

Entradas

Condiciones

- Equipos conformados

Productos de trabajo

- *Planteamiento de Necesidades*
- *Acuerdos del equipo (de todos los equipos)*
- **Herramienta(s) colaborativa(s)** inicializada (s)

PA1 Planificar el proyecto

Actividades

1. Comprender los **objetivos**, los requerimientos **funcionales** básicos y **los requerimientos no funcionales** del software, así como las **restricciones** del proyecto.
2. Conocer y aceptar la **forma de trabajar** para el proyecto.
3. Conocer las **fechas de compromisos** con el cliente y las **condiciones de entrega**.
4. Documentar elementos del **plan del proyecto**.

TÉCNICAS PARA REALIZAR LAS ACTIVIDADES DE PA1

Actividad 1 Comprender los **objetivos**, los **requerimientos funcionales** básicos y los **requerimientos no funcionales** del software, así como las **restricciones** del proyecto. **(1/6)**

- Para identificar y comprender los **objetivos**, los **requerimientos funcionales** básicos y los **requerimientos no funcionales** del software, así como las **restricciones** del proyecto, los **equipos** vuelven a revisar con mayor detenimiento el **Planteamiento de Necesidades** para el proyecto.

Actividad 1 Comprender los **objetivos**, los **requerimientos funcionales** básicos y **los requerimientos no funcionales** del software, así como las **restricciones** del proyecto. **(2/6)**

El primer paso para crear el **Diagrama general de casos de uso** es el siguiente:

1. Cada equipo, **por separado**, hará su **propuesta del diagrama** bajo la coordinación del **Responsable técnico**.

Actividad 1 Comprender los **objetivos**, los **requerimientos funcionales** básicos y los **requerimientos no funcionales** del software, así como las **restricciones** del proyecto. **(3/6)**

- Para identificar a los **actores** y los **casos de uso** se sugiere:
 - Para identificar **actores** se revisa el **Planteamiento de Necesidades** buscando qué **tipo** de diferentes **usuarios** interactuarán con el software.
 - Para identificar los **casos de uso** se busca en el **Planteamiento de Necesidades**, por **cada actor**, qué es lo que podrá **hacer** una vez implementado el software.

Actividad 1 Comprender los **objetivos**, los **requisitos funcionales** básicos y los **requisitos no funcionales** del software, así como las **restricciones** del proyecto. **(4/6)**

- **Hacer el Diagrama general de casos de uso**
 - Para cada **actor** identificado se dibuja el símbolo del actor y se le asigna el **nombre genérico en singular**.
 - Para cada **caso de uso** identificado se dibuja un óvalo y se le asigna el nombre que inicia con un **verbo en infinitivo**, tal como fue identificado en el *Planteamiento del problema*.
 - Para cada **actor** dibujar una **línea recta** entre **el actor y los casos de uso** que puede ejecutar. Un actor puede ejecutar muchos casos de uso y un caso de uso puede ser efectuado por muchos actores.
 - **Todos los casos de uso** identificados **se enmarcan** en un **rectángulo** para delimitar el **alcance** del sistema.

Actividad 1 Comprender los **objetivos**, los **requisitos funcionales** básicos y **los requisitos no funcionales** del software, así como las **restricciones** del proyecto. **(5/6)**

Los siguientes pasos son:

2. El responsable técnico de cada equipo **presenta su propuesta** ante los demás equipos y el docente.
3. Bajo la **coordinación del docente** se **analizan las propuestas** y se genera una de **común acuerdo**. Un voluntario documenta la versión final del **diagrama general de casos de uso** y la **resguarda** en el **repositorio común** del proyecto.

Actividad 1 Comprender los **objetivos**, los **requisitos funcionales** básicos y **los requisitos no funcionales** del software, así como las **restricciones** del proyecto.
(6/6)

- A partir del **Diagrama general de casos de uso generado por** los docentes crean la **lista priorizada** de casos de uso (LPRP), la comparten con los equipos.

Actividad 2 Conocer y aceptar la forma de trabajar para el proyecto.

- En el caso de nuestro curso la forma de trabajar acordada son las prácticas del método MIDS que van a ir aprendiendo y aplicando a lo largo del curso.
- Para coordinar el trabajo de todos los equipos se creará un Grupo de integración conformado por los Responsables de equipo de cada equipo.

Actividad 3 Conocer las **fechas de compromisos** con el cliente y las **condiciones de entrega**.

- Para conocer las **fechas de compromisos** con el cliente, que en nuestro caso son los docentes, y las **condiciones de entrega** el equipo revisa la sección correspondiente del **Planteamiento de Necesidades** y aclara dudas con los docentes.

Actividad 4 Documentar elementos del plan del proyecto.

- Para que todos los equipos tengan **siempre disponible la información importante** del proyecto ésta se documenta usando la plantilla *Plan del proyecto*.
- El **Grupo de integración** (conformado por los **Responsables de equipo**) determina cual de sus integrantes **documentará el plan del proyecto** (usando la plantilla) y lo **resguardará** en el repositorio compartido del proyecto.
- Los **Resposables de calidad** de **todos los equipos** recopilan y proporcionan al Grupo de integración la información de sus equipos y **aseguran la consistencia del plan**.

PA1 Planificar el proyecto

Resultados

Condiciones

- Todos conocen y entienden los elementos del plan del proyecto.

Productos de trabajo

- Herramienta(s) colaborativa(s)
- Plan del proyecto documentado y resguardado

PRACTICA ADMINISTRATIVA RECURRENTE PARA PLANIFICAR LA ITERACIÓN

PA2 Planificar la iteración

Objetivo

- Definir el alcance y prepararse para el trabajo durante la iteración.

PA2 Planificar la iteración

Entradas

Condiciones

- Equipos conformados

Productos de trabajo

- *Planteamiento de Necesidades*
- *Acuerdos del equipo (de todos los equipos)*
- *Plan del proyecto*
- Herramienta(s) colaborativa(s) inicializada (s)

PA2 Planificar la iteración

Actividades

1. Definir el alcance de la iteración.
2. Construir el tablero de la iteración.
3. Crear la carpeta de la iteración en el repositorio común para el resguardo de documentos de la iteración.

TÉCNICAS PARA REALIZAR LAS ACTIVIDADES DE PA2

Actividad 1 Definir el alcance de la iteración.

- Para seleccionar los casos de uso que se incluirán en el alcance de la iteración se aplican los siguientes criterios:
 - Casos de uso que son **prioritarios** para el **cliente** (usando **LPRP**)
 - Casos de uso **independientes** de otros
 - Casos de uso que **dependen de los que ya están implementados**.
- El equipo **selecciona**, de **común acuerdo** con el **docente/cliente** los casos de uso que va a tener que implementar en la iteración.

Actividad 2 Construir el tablero de la iteración. (1/3)

- Para crear el *tablero Kanban* para cada iteración se selecciona una herramienta que ayuda a automatizarlo y hacerlo disponible a todo el equipo.
 - El **Responsable de la colaboración** asegura la **creación** del *Tablero* para el equipo.
 - El **Responsable de la colaboración coloca** en la lista de asuntos (*Issues*) del *Tablero* los casos de uso en el **alcance** de la iteración.

Tablero Kanban del curso

This screenshot shows the GitLab interface for a project named 'espacio-equip-1'. The top navigation bar includes the GitLab logo, project navigation links (Projects, Groups, Activity, Milestones, Snippets), a search bar, and user profile icons. The left sidebar shows the project structure with 'Issues' selected and a count of 1. The main area displays the 'Issue Boards' for 'Administrador > espacio-equip-1'. A search bar and 'Add list'/'Add issues' buttons are at the top. The Kanban board has three columns: 'To Do' (orange header), 'Doing' (green header), and 'RQA' (blue header). Each column has a trash icon, a counter showing '0', and a plus icon for adding new issues.

This screenshot shows the same GitLab interface for 'espacio-equip-1'. The Kanban board now has three columns: an empty column with a blue header, 'RCL' (red header), and 'Closed' (grey header). Each column has a trash icon, a counter showing '0', and a plus icon. The 'Closed' column has a dropdown arrow next to its title. The 'Add list' and 'Add issues' buttons remain at the top of the board area.

Actividad 2 Construir el tablero de la iteración. (2/3)

- El equipo acuerda quién va a ser responsable de implementar cada caso de uso.
- En la columna Por hacer (ToDo) se colocan las primeras tarjetas con las tareas que cada miembro del equipo se compromete hacer con las fechas de terminación.
- A lo largo de la iteración se van agregando las siguientes tarjetas, que permitan lograr la implementación de todos los casos de uso en el alcance de la iteración.

Actividad 2 Construir el tablero de la iteración. (3/3)

- Cuando alguien empieza su tarea mueve la tarjeta a la **columna Haciendo (Doing)** y cuando termina la mueve a la **columna RQA (Review Quality Assurance)** para que el Responsable de calidad revise el resultado de trabajo.
- Una vez revisado, el Responsable de calidad mueve la tarjeta a la **columna RCLI (Review Client)** para que sea **calificado** por el **docente**.
- Finalmente la tarjeta se mueve a la **columna Hecho (Closed)**.

Actividad 3 Crear la carpeta de la iteración en el repositorio común para el resguardo de documentos de la iteración.

- Al inicio de cada iteración se coloca una carpeta nueva en el repositorio compartido, con el identificador de la iteración.
- Esta carpeta servirá para resguardar todos los documentos generados durante la iteración.
- Si se trata de la primera iteración, se colocan en esa carpeta, las plantillas para todos los entregables.

PA2 Planificar la iteración

Resultados

Condiciones

- Todos conocen y entienden el alcance de la iteración y sus compromisos.

Productos de trabajo

- Herramienta(s) colaborativa(s)
- Tablero de la iteración creado para el alcance de la iteración
- Herramienta(s) colaborativa(s) que incluye una carpeta para la iteración

PRACTICA ADMINISTRATIVA RECURRENTE PARA EJECUTAR EL PLAN, DAR SEGUIMIENTO Y CONTROLAR LA ITERACIÓN

PA3 Ejecutar el plan, dar seguimiento y controlar la iteración

Objetivo

- Visibilizar el avance de las actividades del proyecto y aplicar acciones correctivas oportunamente.

PA3 Ejecutar el plan, dar seguimiento y controlar la iteración

Entradas

Condiciones

Todos conocen y entienden el alcance de la iteración y sus compromisos.

Productos de trabajo

- Herramienta(s) colaborativa(s)
- Tablero de la iteración creado para el alcance de la iteración
- Herramienta(s) colaborativa(s) que incluye una carpeta para la iteración

PA3 Ejecutar el plan, dar seguimiento y controlar la iteración

Actividades

1. **Ejecutar** lo planeado para la iteración y darle seguimiento revisando y actualizando el **tablero**.
2. **Evaluar** el estado del **progreso** de la iteración a través del tablero, identificando **dificultades y retrasos**.
3. Proponer **soluciones** para **controlar** las dificultades y actualizar el tablero con las **actividades acordadas**.

Actividad 1 Ejecutar lo planeado para la iteración y darle seguimiento revisando y actualizando el **tablero**. (1/2)

- Los **miembros del equipo** al iniciar las **actividades planificadas** en la columna **Por hacer** del tablero , mueven la tarjeta a la columna **Haciendo** y cuando terminan la mueven a **Hecho**.
- Durante las **reuniones diarias** se da el **seguimiento** a los compromisos y se **acuerdan nuevos**, que deben reflejarse en el **tablero**.

Actividad 1 Ejecutar lo planeado para la iteración y darle seguimiento revisando y actualizando el **tablero**. (2/2)

- En la **reunión diaria del equipo**, que no debe durar más que 15 min, todos contestan 3 preguntas:
 - ¿Qué he hecho desde la última reunión?
 - ¿Qué haré para la próxima reunión?
 - ¿Qué dificultades he encontrado?
- El **Responsable de la colaboración** se asegura que **todos actualicen** el *Tablero* moviendo sus tarjetas según el estado del trabajo de cada día.

Actividad 2 Evaluar el estado del **progreso** de la iteración a través del tablero, identificando dificultades y retrasos.

- Durante las **reuniones diarias** se pueden identificar **dificultades y/o retrasos**.
- En el caso de que no sean de fácil solución, se procede a realizar la **Actividad 3** de esta práctica.

Actividad 3 Proponer soluciones para controlar las dificultades y actualizar el tablero con las actividades acordadas.

- En caso de dificultades o retrasos, un miembro del equipo convoca a la reunión de toma de decisiones siguiendo la práctica emergente PS3.
- Todo el equipo analiza las causas y propone acciones para remediar los problemas.
- Se acuerdan responsables de las nuevas actividades y se incorporan al tablero como cosas Por hacer.
- Si el equipo no puede resolver el problema se avisa de manera inmediata al docente.

PA3 Ejecutar el plan, dar seguimiento y controlar la iteración

Resultados

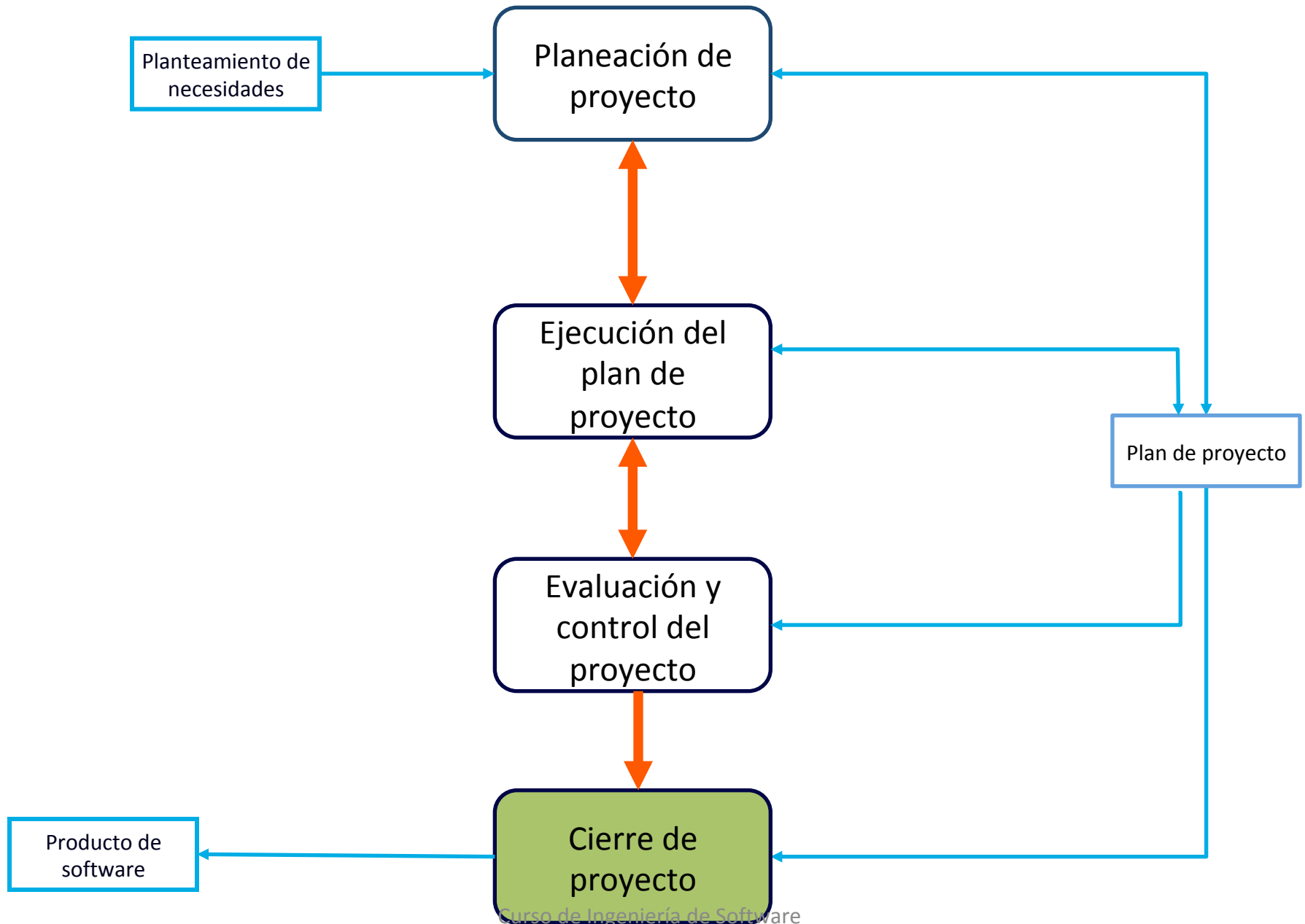
Condiciones

- Todos saben como ejecutar el plan de la iteración, dar el seguimiento usando el tablero y comprenden cómo resolver las dificultades.

Productos de trabajo

- Herramienta(s) colaborativa(s)

Actividades del proceso de administración del proyecto



Cierre de la iteración y del proyecto

- Las actividades administrativas de cierre de la iteración y del proyecto se presentarán en la Unidad 9, cuando terminaremos la iteración y, posteriormente, el proyecto.

Resultados de esta unidad

Condiciones

- Todos conocen y entienden los elementos del plan del proyecto.
- Todos conocen y entienden el alcance de la iteración y sus compromisos.
- Todos saben como **ejecutar el plan de la iteración**, dar el **seguimiento usando el tablero** y comprenden cómo **resolver las dificultades**.

Productos de trabajo

- **Herramienta(s) colaborativa(s)** que incluye una **carpeta para la iteración**
- **Plan del proyecto** documentado y resguardado
- **Tablero de la iteración** creado para el alcance de la iteración

Referencias

- Anderson, D. J. (2010). *Kanban. Successful evolutionary change for your technology business*. Blue Hole Press.
- ISO/IEC 29110. (2011). *29110-5-1-2Software Engineering-lifecycle Profiles for Very Small Entities Management and Engineering Guide. s.1. Software Engineering*.
- Jacobson I. (2012). *Uses Case 2.0*.
- ISO/IEC 25010. (Dic. 2009). *Systems and software engineering -System and software Quality Requirements and Evaluation (SQUaRE)-System and software quality models*.
- PMBOK. (2013). *Project Mnagement Body of Kowledged*. PMI.
- Rosenberg Doug, S. K. (2001). *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*. Addison Wesley.
- Rumbaugh J., I. J. (1998). *The Unified Modeling Language. Reference Manual*. Addison Wesley.
- SEVOCAB . (2017, 08 11). Retrieved from Software and System Engineering Vocabulary: https://pascal.computer.org/sev_display/index.action
- SWEBOK 3.0. (2014). *Guide to the Software Engineering Body of Knowledge v3.0*. IEEE Computer Society.