

Toma de Notas 7/9

Proceso, Proyecto, Producto Personas son los ejes sobre los que la disciplina de ingeniería e software trabajaba, son los ejes, las dimensiones de los que la ingeniería de software se ocupa, se ocupa del producto como objetivo final, entregar un producto de software de calidad al cliente es la motivación fundamental y el concepto de proyecto es como se dijo por muchos años el medio por el que yo obtengo un producto de software.

El proyecto es la unidad organizativa, la unidad de gestión de la disciplina. El proyecto para poder existir necesita de una serie de elementos, necesita de personas, de recursos y necesita de un método, de una forma, una indicación de cómo hacerse el trabajo que hay que hacer para cumplir con el objetivo que este proyecto se plantea que en este caso es tener un producto de software.

Ahí es donde entra a jugar el proceso porque el proceso es el que da una definición teórica de qué es lo que debería hacerse para hacer software. El proyecto debe tomar de esa definición teórica de acuerdo conforme lo necesite debe tomar de esa definición de proceso lo que le parece que hace falta, lo que es mejor con un criterio. No con una cuestión de me gusta esto lo hago, no me gusta no lo hago sino con un criterio de necesidad y de corrección y de adecuación toma de esa definición teórica lo que le hace falta y arma el conjunto de tareas que es necesario hacer y ese conjunto conforma el alcance del proyecto porque el alcance del proyecto se define como el trabajo que hay que hacer para cumplir con el objetivo.

Entonces ese alcance del proyecto que son tareas o actividades se toman de esta definición teórica de proceso. Es importante que nosotros tengamos esta distinción conceptual en nuestra cabeza porque si usamos los términos indistintamente esta cuestión de hablar del proyecto como si fuera el proceso o viceversa es común y lo mismo pasa con la mezcla que se hace entre proyecto y producto que no son lo mismo.

ISW cambia el foco, trabaja en proyecto y vamos a trabajar en proceso como eje. Análisis y diseño hablan de proceso pero hablan de disciplina del proceso que tiene que ver con la ingeniería del producto, hablamos de requerimiento, análisis, diseño. NO habla de disciplinas de cómo gestionar proyectos.

Hay dos grandes tipos de procesos, los empíricos y definidos. Lo que pasa con los empíricos es que no hay ninguno que sea un proceso completo. El PUD intenta cubrir todo lo que haga falta para producir software, un SCRUM, un XP, todos estos frameworks o metodologías no se les dice procesos porque no están completos, son lineamientos, son buenas prácticas de alguna parte o partes de un proceso pero como son empíricos la idea es que el equipo cierre y defina y crea su propio proceso para el proyecto entonces ninguno de estos es un proceso completo. Por eso no es correcto decir que es SCRUM es un proceso, es un framework de trabajo para la gestión de productos y proyectos de cualquier cosa.

Proceso definición de un conjunto de actividades que toman como entrada requerimientos y obtienen como salida un producto o servicio, en este caso de software. Ese conjunto de actividades están estructuradas y guiadas por un objetivo que es lo que quiero conseguir con ese proceso, eso es un proceso. Definición teórica de lo que hay que hacer para hacer un software, primero hablar con el cliente, preguntarle que necesite, elicitar los requerimientos, hacer análisis, diseño, arquitectura.

PUD describe todas las cosas que tienes que hacer organizadas por flujos de trabajo para poder obtener el producto final pero es una teoría, lo que deberíamos hacer para hacer software, eso es el proceso.

El proyecto es una unidad de gestión, es un medio por el cual yo administro los recursos que necesito y las personas que van a estar integradas y trabajando, medios de gestión de organización para obtener como resultado un producto o servicio, que en nuestro caso es software.

Entonces el proyecto empieza decidiendo y eligiendo la gente que va a sumarse a trabajar o los distintos roles en el proyecto. Incorpora personas que van a asumir roles, esos roles teóricos están definidos en el proceso, entonces el PUD define un rol de arquitecto entonces buscamos los que tenemos disponible y decimos en este proyecto necesito un arquitecto, otro funciona como ingeniero de casos de uso, otro como líder de proyecto. Entonces ya tenemos el equipo con sus roles, también puede pasar que el arquitecto sea además diseñador de interfaces porque una persona puede asumir mas de un rol en un proyecto. También puede pasar que haya un rol como desarrollador, ingeniero de componentes que lo asuman muchas personas, eso también es común, que haya mas de un desarrollador o más de una analista en un proyecto. Cuando sumo la gente esta es la gente que va a ser la responsable de hacer todo el trabajo que haya que hacer para obtener el producto que queremos obtener.

El libro usa un termino de se instancia, el proceso se instancia en el proyecto porque el proyecto le da vida al proceso, el proceso dice teóricamente lo que hay que hacer pero el proyecto define las tareas basadas en ese proceso diciendo que esta tarea me sirve la hago, no me sirve no la hago y arma el conjunto de tareas que se van a llevar adelante en el proyecto. Las tareas se ejecutan concretamente en el ámbito del proyecto porque el proceso es como buscar en el diccionario una definición, es una teoría.

Producto es el software, que es software? Que entendemos por software? La mayoría asociamos un aplicativo o un sistema o un programa funcionando en una computadora. Software es conocimiento empaquetado a distintos niveles de abstracción. Es cierto que el código instalado en una computadora funcionando es software, pero la definición de producto también es software, una base de datos también es software, una user story también es software, el diseño entre una interacción del usuario y el sistema también es software. Tenemos que tener una visión mas amplia del concepto porque si solo hacemos foco a que software es el código ejecutable estamos perdiendo un montón de información que es la que me permite a mi llegar a tener ese ejecutable. Cuando nosotros hablamos de proyectos que genera producto cada tarea del proyecto debe tener un entregable porque si no tenemos un entregable concreto no sabemos si la tarea se hizo o no se hizo. Entonces las primeras tareas van a tener como entregables requerimientos diseños modelos, diseños de bases de datos, diseños de arquitectura y las tareas relacionadas a la implementación van a tener como entregable el código fuente que después será código a ejecutar.

El producto de software incluye todo, e manual de usuario que le doy al cliente son software el manual de configuración es software. Todo lo que sale como resultado de ejecución de la tarea de un proyecto es software. Ese es el enfoque que le damos en esta disciplina. Cada artefacto que sale de una tarea que se realiza en un contexto de un proyecto es software que en términos de la disciplina de gestión de configuración tiene un nombre, cada pieza lo nombra como ítem de configuración. Cada pieza que la gestión quiere controlar y mantener integra y versionar se llama ítem de configuración. Cada file system que guardamos en un disco es un ítem de configuración y para nosotros es software.

Hay una corriente ahora que se sumo de la mano de la innovación y de toda esta onda de startups que esta intentando introducir en nuestras mentes el concepto de producto y fortalecer el concepto de producto y entender que el proyecto es el medio para obtener una versión del producto. El ciclo de vida del producto es mas grande porque va a vivir mucho mas que el proyecto que lo crea, el concepto para lo que es software es software producto. Producto que es lo que finalmente le entrego a mi cliente.

Que es un proyecto

Hay un producto resultante, es único y tiene un fin. Esto empieza y termina. Ese es el concepto de temporalidad que hace que tengamos que evaluar si algo califica como proyecto o no. Los proyectos tienen características muy específicas que son las que vamos a ver ahora.

Los proyectos deben tener un objetivo único, esto significa que es distinto de los productos resultantes de otros proyectos, cada resultado de un proyecto es distinto del resultado de otro proyecto. Una mamá puede tener 20 chicos pero cada uno es distinto. Cada vez que hacemos software cada uno de ellos es distinto del anterior y distinto del siguiente aunque sea el mismo producto porque van a ser versiones distintas del producto que tienen distintas cosas que hacer. Cada versión que se libera de un producto de software es distinta al anterior, de esa manera reforzamos que el resultado de un proyecto es único.

Para obtener y lograr ese objetivo que tiene que ser único el proyecto está guiado por un objetivo que hay que cumplir que es obtener ese producto único. Ese objetivo tiene que cumplir con dos restricciones, características, muy importantes. Debe ser claro para todos los que están involucrados y debe ser alcanzable. Que un objetivo sea alcanzable significa que tenemos que sentir que si le ponemos lo que le tenemos que poner lo podemos lograr. Si participamos de proyectos que tienen objetivos no alcanzables estamos participando de un proyecto condenado al fracaso. Yo puedo plantear un objetivo claro. El objetivo del proyecto está relacionado con el objetivo del producto pero no es lo mismo, el objetivo del proyecto define el trabajo que el proyecto tiene que hacer para dejar contento al cliente y a los demás involucrados.

Después tenemos esta cuestión temporal, el proyecto es algo de flujo constante, no es una línea de fabricación en la que vamos entrando materia prima y va saliendo producto. El proyecto empieza y termina, tiene una fecha de inicio y una fecha de fin, si o si. En algún punto el proyecto termina, liberamos los recursos (mandamos a la gente a otro proyecto) y quedará asignado con el mismo u otro rol en otro proyecto. A eso nos referimos con la liberación de los recursos. Desafectamos el espacio de trabajo, recursos de software, las licencias y las personas para que estén en condiciones de trabajar en otro proyecto, ese es el concepto de recursos libres.

Que pasa con la manera en la que alcanzamos el objetivo, es esto que hablamos acá de tareas interrelacionadas. Yo tengo un objetivo que es claro, que es alcanzable, tengo la gente que va a trabajar, los recursos que afecte y ese elefante me lo como en pedazos, el famoso enfoque de divide y vencerás. Dividimos todo el trabajo que debemos hacer en tareas que tienen que tener una relación en términos de que hay tareas que se pueden hacer en paralelo, otras no, hay tareas que tienen que estar terminadas para empezar otras tareas y a eso nos referimos con la interrelación que genera dependencias inevitables porque estamos trabajando en un sistema que es interrelacionado. Armamos ese conjunto de tareas donde la definición de que tareas tenemos que hacer las sacamos del proceso.

Entonces las sacamos del proceso para no tener que pensarlas de cero. Lo que es único tenemos que si o si entender que el resultado de esto es único cada vez.

Entonces las características son único, fecha de inicio y fin, objetivo que tiene que ser claro y alcanzable y las tareas interrelacionadas. Esas son las 4 características que debe tener un proyecto para considerarse como tal.

Proyectos se planifican o no?

Yo puedo tener un proyecto debería estar planificados, porque a veces no se planifica. En algún punto puede ser que no se empiece planificando pero en algún punto lo tenes que planificar porque un proyecto planificado fracasa a veces, imagínense si no se planifica para nada.

Lo importante de la planificación es el acto de planificar, no el resultado y ahí es donde resulta que la industria esto de la planificación tiene mala prensa porque mucha gente toma esto como una responsabilidad de completar un template de un plan de proyecto en el que alguna cosa tiene que escribir en cada casillero del template y ni lo piensa. Si esa es la actitud no sirve, lo que sirve es el acto de planificar, el acto de ponerme en el ejercicio de pensar cual es objetivo del proyecto, para que lo hacemos, cual es el alcance y que recursos necesitamos, hay algo que nos puede poner en peligro el objetivo del proyecto eso se llama riesgos. La acción de hacer ese trabajo es lo valioso, de eso aprendemos y después decidimos como equipo que tanta formalidad queremos dejar guardada o registrada o documentada (a mí no le gusta usarla mucho porque a la gente le da alergia), la cuestión no es documentar por documentar sino decidir como equipo de que cosas es importante dejar registro, que decisiones tomamos en este espacio de planificación que estaría bueno que quedaran escritas de alguna manera para que no nos tengamos que andar acordando de tantas cosas y segundo porque si necesitamos comunicárselo a alguien está ahí y tercero porque tenemos más probabilidad de lograr algo que no es fácil de lograr que es una visión compartida, es decir, que todos entendamos lo mismo sobre algo es más fácil cuando vos lo escribís, lo dejás escrito de alguna manera en algún lado. De lo contrario cada uno de nosotros tiene una idea en su mente pensando que es la misma que no siempre lo es o que la mayoría de veces no lo es. Esa es la ventaja de dejar algún registro de estos procesos de toma de decisiones, de arquitectura, de requerimientos, de cómo nos manejamos en el código, eso nos significa que tiene que un libro de 50 páginas puede ser 2 carillas, puede ser un afiche pegado en la pared. Puede ser una tarjeta en una herramienta como trello, no importa el medio, lo importante es que esas decisiones respecto de los planes queden descritas en algún lugar. Lo importante es lo que uno aprende en el proceso, no el resultado.

¿Qué es la administración de proyectos?

Básicamente hablamos de que quiero que este proyecto cumpla con el objetivo exitosamente entonces voy a administrar todos los recursos que me dieron y voy a organizar el trabajo de la gente afectada y voy a hacer un seguimiento de si las cosas se están dando como nosotros dijimos que se iban a dar y finalmente todas esas actividades que intenta controlar y llegar a buen puerto con el objetivo del proyecto eso se conoce como administrar un proyecto.

Yo tengo gente, tengo recursos, tengo un objetivo, si lo dejo no hay probabilidades de que salga bien. En este enfoque de gestión tradicional hay una figura que se llama líder de proyecto que es el responsable de hacer este trabajo, de administrar el proyecto. La administración de proyectos tiene dos grandes actividades, una es la planificación y la otra es el seguimiento y el monitoreo de que las cosas pasen como las planifique.

La administración de proyectos implica estas dos disciplinas, la planificación y el seguimiento y control, esas dos juntas es a lo que nos referimos cuando hablamos de administrar un proyecto.

Aquí es cuando surgen distintos enfoques para administrar un proyecto. En esta materia el foco es la administración de proyectos ágiles pero para llegar a entenderlo necesitamos primero ver el enfoque tradicional porque lo ágil se define por comparación contra esto otro, entonces si no vemos mínimamente algo de esto otro no vamos a poder entender lo que ágil dice.

Triple restricción

Las 3 dimensiones a cumplir por un equipo de sistemas es por un lado el producto que se tiene que entregar que es la que se refiere como alcance, que son los requerimientos que el cliente me expresó y que yo de alguna manera entendí. Para lograr entregar ese producto o servicio necesito tiempo y recursos que terminan generando un costo (recursos y gente). En el software el costo más significativo de un proyecto es el costo del trabajo de la gente, el 80% del costo de un proyecto de software son los honorarios de la gente que trabaja en el.

Si quiero ser líder de proyecto dos de estas la tenemos que poder controlar, no entremos a trabajar en un proyecto donde las 3 dimensiones estén fijas porque no tenemos margen de maniobra. La dimensión que normalmente no controlamos nunca es el alcance porque el producto es del cliente, el que lo quiere y necesita es el cliente, el que te va a decir que es lo que quiere es el cliente y el que te paga si queda conforme es el cliente. El alcance no lo manejamos nosotros pero dado este alcance déjame que yo te diga que recursos necesito, cuanto tiempo necesito y eso cuanto plata me va a salir. Las patas de abajo son las que nosotros podemos determinar.

Si el cliente cambia el alcance las otras dos dimensiones cambian por consecuencia, el problema eterno que tenemos con la profesión es que la definición de alcance en un proyecto se hace a riesgo normalmente. Cuando empezas en un proyecto un cliente te convoca, te cuenta el software que quiere y si después el cliente te dice que no lo quiere todo el esfuerzo en determinar esos requerimientos no lo cobras lamentablemente. Esa es la razón por la que no queremos invertir tanto en entender el alcance pero eso es una trampa porque nos comprometemos con algo que termina no siendo lo que el cliente después pidió porque en el medio te agrega cosas y como vos ya le diste un tiempo y un costo no te deja que lo cambies o no es tan fácil cambiarlo.

De esa situación es que nace la resistencia tan terrible de los cambios en los requerimientos, porque si cambian los requerimientos cambian el tiempo y el costo y el cliente no te lo quiere pagar. El problema es que no te dan mas tiempo y no te lo pagan.

Entonces no es que haya un enfoque que es bueno y el otro malo, es como nosotros encaremos, negociemos y que cliente tenemos de otro lado, si el cliente es razonable esto debería funcionar bien. La moraleja del cuento es que las tres dimensiones están relacionadas y que si nosotros vamos a hacer un enfoque de gestión basado en el alcance, el alcance determina el tiempo y el costo, cambia el alcance y lo otro va a cambiar.

La mayoría de los casos tenemos que poder tener control por lo menos de estas dos dimensiones porque si estos se desequilibran lo que se sacrifica es la calidad. Definimos con el cliente que queremos un fiat 600 y se lo vamos a entregar en tantos meses con tanta planta, en el medio el cliente fue cambiando de requerimiento y terminó con la expectativa de que sea un mercedes benz y no quiere que le cambies el tiempo ni el costo y como nosotros no somos capaces de decirle que no sea viable entregamos la carcasa del mercedes benz pero adentro la ingeniería, la mecánica, el motor es del fiat 600 porque es lo que llegamos, afectando la calidad del producto. Eso nunca debería ser en esos términos, vos lo que negocias es el alcance pero no ajustas la calidad del producto. La situación normal en las empresas es que te entreguen un producto sin testing o sin haber hecho sin revisiones de calidad o sin ver el código por dentro, o no se pensó la arquitectura y después no se entiende porque anda mal el software.

Hay cosas que no debería ser negociables, testing no es algo que te pones a pensar que si tenes tiempo haces testing. Vos cuando vas al medico no le decís como te tiene que operar, operame sin cerrar la herida, eso es lo mismo que entregar un software sin testing, El software como es intangible

es difícil imaginarse que pasa cuando entregas un producto sin hacer testing pero el efecto a la corta o a la larga es mas o menos lo mismo.

La calidad se afecta si esto no esta equilibrado, la calidad de producto, lo que le entregamos al cliente.

Rol del líder de proyecto / equipo

En esta gestión tradicional tenemos este concepto de equipo donde hay un líder que se relaciona con los demás involucrados y es generalmente bastante conductistas, le dice a la gente lo que tiene que hacer, cuando lo tiene que hacer, para cuando tiene que estar terminado y el equipo recibe las asignaciones y trabaja de la mejor manera posible informa cuando termino, el porcentaje de avance y si hay una desviación interesante respecto a los planes debería verse que acciones correctiva tomar en base al desajuste. Eso es el estilo de gestión tradicional.

Como lideres tenemos la responsabilidad de estimar, de planificar, de asignar trabajo a la gente y de hacer seguimiento de lo que pasa con la gente en cada momento que va pasando del proyecto. El trabajo de un líder de proyecto es este, no suele tener trabajo técnico, como programar hacer requerimientos. En estos enfoques eso no se recomienda porque si la misma persona hace trabajo técnico y de gestión la mayoría de las veces lo que pasa es que prioriza el trabajo técnico por sobre el de gestión entonces el proyecto no lo gestiona nadie.

Por eso la recomendación de estos enfoques es que el líder de proyecto gestiona el proyecto, no tiene asignaciones de programación porque sino no gestiona nadie.

Plan de proyecto

Tenemos que obtener como lideres el plan de proyecto y es el plan de proyecto el que de alguna manera nos va a guiar a nosotros en nuestro viaje, es como si fuera un mapa que define todo lo que queremos hacer, como lo vamos a hacer, cuando lo vamos a hacer. Este es el artefacto resultante del proceso de planificación. Lo importante es el fuerza que le pusimos al plan, no tanto el plan.

Tenemos que responder la siguientes preguntas en un plan de proyecto:

¿Qué es lo que hacemos? Seria el alcance

¿Cuándo lo vamos a hacer? Eso es la calendarización

¿Cómo lo hacemos? Seria la afectación de recursos y las decisiones de las tareas

¿Quién lo hace? Es la asignación de las personas.

¿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

Esto es planificar un proyecto de software con la gestión tradicional, tenemos que hacer todo esto. Tenemos que definir el alcance del proyecto, elegir un proceso y un ciclo de vida, estimar, identificar riesgos, asignar recursos, la programación del proyecto es la calendarización que es decir quien hace que a que hora. Después definimos como vamos a controlar el proyecto, es decir, voy a hacer tal reunión con el cliente tal día, tal reunión con el equipo tantas veces a la semana, voy a generar estos informes y voy a usar estas métricas para ver como me va yendo. Si lo que planifique esta siendo como lo planifique o no y ahí es cuando aparecen las métricas. Las métricas surgen porque son un numero y un numero es una definición cuantitativa, es bueno esto porque los números son objetivos porque sino decimos que nos va bien, nos falta poco pero no te deja claro si vamos bien o falta poco ya que es una percepción subjetiva de la realidad, te pueden estar diciendo la verdad, te pueden estar mintiendo, te pueden decir lo que ellos creen que es la verdad que en realidad no lo es porque es una percepción de la realidad que el equipo tiene. EN cambio si vos lo medis se vuelve objetivo, si terminaste todas las tareas sabes que vas bien por ejemplo. EL tiempo corre pero que vos sumes tiempo no significa que el trabajo este hecho. Necesitamos las métricas porque las métricas son objetivas, las métricas son la realidad, lo demás es percepción.

Después cuan formal sea o no el artefacto resultante es otra cosa. El como es un detalle, lo importante es que tenga respuesta para todo lo planteado acá.

Definición del alcance

La primera confusión con esto es la diferencia entre el alcance del producto y del proyecto. Producto tiene un alcance que es la sumatoria de todos los requerimientos funcionales y no funcionales que ese producto tiene que satisfacer, eso va guardado en la ERS.

El alcance del proyecto, trabajo, tareas, las tareas que tengo que hacer para cumplir con el objetivo del proyecto. El alcance lo guardo en el plan de proyecto.

La relación entre estos alcances es que si el producto que tengo que construir es mas grande las tareas del proyecto van a ser mas o van a necesitar más tiempo o van a necesitar mas gente. Yo para definir el alcance del proyecto tengo que tener definido antes el alcance del producto. Por eso se relacionan pero el alcance del producto se mide en requerimientos, el alcance de proyecto se mide en tareas, esa es la diferencia. El alcance del producto va en la ERS y el del proyecto va en el plan de proyecto. Esos son los entregables donde guardamos cada uno de estos dos, si hay relación pero no es lo mismo.

Definir un ciclo de vida

El ciclo de vida no es lo mismo que el proceso, el proceso define las tareas que hay que hacer, el ciclo te dice como vas a hacer esas tareas, cuanto de cada tareas vas a hacer cada vez. Lo importante en el contexto de la planificación de proyecto es saber que cuando incia el proyecto tengo que elegir que proceso quiero usar y que ciclo de vida quiero usar, el proceso unificado te recomienda este ciclo de vida que esta aca que es el ciclo de vida iterativo e incremental pero si alguien quiere usar el proceso unificado pero con ciclo de vida en cascada se puede hacer.

En estos enfoques de gestión tradicional puedo elegir el proceso y elegir cualquier tipo de ciclo de vida para llevar adelante el proceso, no hay ninguna restricción, en ágil es otra historia, te condicionan, si quieres trabajar con gestion gil hay un solo ciclo de vida que podés usar que es el iterativo, no hay otra opción. La combinatoria gestión ágil con ciclos de vidas secuenciales, ni tampoco con cascada. La gestion gil se basa en procesos empíricos que son procesos que se alimentan con la experiencia, no podes alimentarte de ninguna manera con un ciclo de vida en cascada porque la experiencia llega demasiado tarde.

Estimación de software

En la gestión tradicional se plantea una secuencia de estimaciones que tiene que tener este orden, lo primero que se estima es el tamaño del producto se software, voy a construir un barquito o un transatlánticos, el tamaño del producto que voy a construir. Cuando estime el tamaño estimo esfuerzo que es las horas personas lineales, es decir, una bolsa de horas que son ideales sin tener en cuenta el tiempo que fueron al baño, horas de trabajo concentrado.

Cuando estime el esfuerzo puede derivar el calendario donde necesito saber la cantidad que trabajan, los días de trabajo, las horas cada día reales. El calendario contesta cuando va a estar esto para poder contestar cuando voy a poder entregar esta versión del producto.

Cuando definí lo anterior recién hago el costeo porque esa es la manera de hacer un costeo mas ajustado a al realidad, el trabajo rápido siempre es mas caro, pero no se puede abusar de eso porque hacemos trabajo intelectual, la gente se cansa. Esto hasta un cierto punto se puede acelerar.

Los recursos críticos son esas cosas raras que nos hacen falta mas allá de lo normal. Si hacemos un software que maneja sensores de alarma de incendio el que desarrolla necesita el sensor porque tiene que desarrollar para ese sensor en particular. El que hace testing también lo necesita para poder testear el sensor. Ese tipo de cosas son recursos críticos, los tengo que identificar, no siempre hay pero si hay los tengo que identificar para ver cuando los voy a tener disponible y cuanto voy a necesitar para que el proyecto no impida su avance por la falta de disponibilidad de esos recursos, a eso lo llamamos recurso critico y también se estima. Se estima la necesidad, cuantos voy a necesitar y cuando los tengo que sumar al proyecto.