

28/3 - Teórico

Slides en UV

Componentes de Proyecto de Desarrollo de Software

Proceso de SW: Conjunto de actividades estructuradas, con el fin de desarrollar un producto de software.

Personas, procedimientos, métodos, herramientas

La org es la que define el proceso, que adopta distintas formas de acuerdo a la organización.

definidos vs empíricos.

Procesos definido: Intenta ser idempotente(mismas entradas dan mismas salidas), inspirado en línea de producción. Asume entradas. Entradas identificadas → un proceso definido → salidas esperadas.

Proceso empírico: Aprender a medida que avanza en la ejecución del proceso. Se busca responder de mejor manera ante la complejidad del ambiente. Solo utilizan modelos iterativos e incrementales. Asume que se aprende de la experiencia, asume procesos complicados con variables cambiantes. Poseen reglas y definiciones, pero en mucho menor medida que los procesos definidos.

Ciclos de vida

Serie de pasos a través de los cuales el producto, proyecto, proceso progresa.

Ciclo de vida del **producto** vs CDV de **proyecto**.

Proyecto: Inicio, fin, cuando termina, tenemos un producto, que hay que mantener. Se puede hacer retiro de un producto.

CDV de Proyecto: Representación del proceso.

Bibliografía: Modelo de proceso.

artefacto: entregable del proceso(vista de CU)

Proyecto

- Objetivo claro, y definido que se debe alcanzar al final del mismo.
- Tiene principio y fin
- Tareas relacionadas entre si, desarrolladas por recursos, mediante el esfuerzo.

Gestion tradicional de proyectos

Cuando se utiliza un proceso definido de desarrollo de software. aplicacion de conocimientos, habilidades y herramientas para lograr el objetivo, y satisfacer los requerimientos del proyecto.

- Restricción triple. Alcance - Tiempo - Dinero
- Como el lider de proyecto juega con la triple restriccion a lo largo de la vida del proyecto, mientras estas varían.
- El Rol de PM se da en la gestion tradicional de proyectos, en gestión ágil (SCRUM/Kanban) no.
- Rol de Lider de Proyecto. completar
- Equipo de proyecto: Conjunto de personas comprometidas a alcanzar un conjunto de objetivos de de los cuales se sienten mutuamente responsables.
 - Diversos conocimientos y habilidades
 - Usualmente pequeño.
 - Desarrollan sinergia.
- [] Leer manifiesto agil.

Plan de Proyecto

- Una hoja de ruta de nuestro viaje
- Define alcance, costos y tiempos.
 - Que hacemos, alcance (proyecto y producto)
 - Cuando, tiempo
 - Como, actividades, recursos, modalidades
 - Quienes, roles
 - Conjuntos de definiciones en filminas.
- **Definicion de alcance de Producto:** Caracteristicas que pueden incluirse en un producto o servicio.
- **Def. de alcance de Proyecto:** Todo el trabajo y solo el trabajo que debe hacerse para entregar el producto/servicio con todas las caracteristicas y funciones especificadas.
- Estimaciones de software: Tamaño, esfuerzo, calendario, costo, recursos críticos.
- **Gestion de riesgos:** identificarlos, ponderarlos (impacto y probabilidad de ocurrencia), se gestionan aquellos con mayor exposición. Se decide acorde a la criticidad del proyecto.
- **Métricas de software:** de proceso, de proyecto(retrabajo por ej) y de producto. Las metricas de proyecto se consolidan para crear métricas de proceso que sean publicas para toda la org. No se usan para medir a las personas. Las metricas de proceso son las mismas que proyecto y nos permiten optimizar los procesos de la org, la de proy mira el contexto de proyecto. Las de producto se relacionan con el tamaño y los defectos.
- Monitoreo y control. Gestionar acorde a los desvíos detectados. Los proyectos se rechazan día a día. Leer Mythical Man Months
- Factores de éxito del proyecto:
 - Monitoreo y feedback
 - Mision/objetivo claro

- Comunicación.

11/04 - Teórico - Estimaciones de SW

La estimación:

Aplicación de distintas técnicas para estimar el tiempo del proyecto. Existen estimaciones de esfuerzo, cronograma, costo, etc. (mejorar)

- No es precisa, componente de incertidumbre. Mientras más se estima, incurrimos en más riesgo.
- Son la base de los planes pero no son lo mismo. Estimar no es planear y planear no es estimar. Al planificar existen cuestiones de negocio/otras que se consideran, fuera de las estimaciones. Ej: Se arma un plan donde se deben cumplir cuestiones contractuales, lanzamiento de producto, cumplir con contextos de licitación. En algunos casos, más allá de los inputs que se tienen, se puede decidir planear distinto a las estimaciones.
- No son compromisos.
- Usando más de una técnica, hay más probabilidades de tener un mayor nivel de certeza.

El problema es estimar inicialmente y no volver a estimar, las estimaciones deben ajustarse a medida que el proyecto avanza, sabiendo que tenemos que asumir incertidumbre.

Errores de estimación: No tener claro, alcance, tamaño del sw a construir.

Unidad de medida: User stories, casos de uso.

Estimar el esfuerzo, cantidad de horas lineales que me lleva hacer el trabajo necesario para construir el producto de SW.

Se calcula el tiempo en horas hábiles. y se toman las horas netas.

- Se pueden tener errores, no tener en claro la capacidad del equipo, subestimar/sobreestimar.
- Se generan errores que se replican en nuestras estimaciones, cuando hay demasiado caos en el proyecto, probablemente se tengan errores en las estimaciones.
- Omitir actividades necesarias para la estimación del proyecto.
- Uso de buffers, no es el colchón. El buffer implica riesgos que generan ese lapso de tiempo que se estima.

Técnicas fundamentales de estimaciones

- Contar: Estimar a través de la cuenta de, req, features, US, CU. Con el avance del proyecto se cuentan otras cosas.

Basados en:

- la experiencia:
 - Datos históricos, proyectos similares, es necesaria una base de conocimiento. Más allá de tener una base de conocimiento, es necesario tener en cuenta las capacidad del equipo que lleva adelante el proyecto, teniendo en cuenta los posibles riesgos, es necesario ajustar estos datos acorde al proyecto.
 - Juicio Experto:
 - Puro: Alguien que es experto, estima. Enfoque más usado.
 - Wideband Delphi: Estimación en grupo, personas informadas tratan de estimar el costo de desarrollo tanto en esfuerzo como en duración. El metodo implica que cada miembro estime por separado. Se repite el proceso hasta que la estimación converge de forma razonable.
- los recursos
- el mercado
- componenets del producto.
- Métodos algorítmicos

Si las estimaciones se utilizan como compromisos son muy peligrosas y perjudiciales para cualquier organización.

Lo mas beneficioso de las estimaciones es el proceso de hacerlas

Sirven como respuesta temprana sobre si el trabajo planificado es factible o no.

La estimación puede servir como una protección para el equipo.

Estimaciones Ágiles: Wideband Delphi.

- Las features son estimadas usando una medida de tamaño relativo, Story Point.
- Las medidas relativas no son absolutas.
- SP no es una medida basada en el tiempo.

Tamaño vs Esfuerzo

Las estimaciones basadas en tiempo son más propensas a errores.

Se toma una feature o US, y se estima cuantos SP conlleva, es relativa al equipo.

Se mide:

- Complejidad
- Esfuerzo
- Incertidumbre

Story Point: Unidad de medida específica (del equipo) de, complejidad, riesgo y esfuerzo. Da la idea de “peso” de cada story, y decide cuan grande es.

No existe una correlacion entre horas-hombre y story points, dado que los mismos 140 sp de una variacion de arquitectura pueden significar un esfuerzo (Hs Hombre) de 6 meses para mi equipo, mientras que los mismos 140 sp, significan 3 meses para otro equipo.

No puedo usar las horas hombre del esfuerzo para apurar a mi equipo? No, porque las estimaciones en tiempo son las mas propensas a errores.

14/04 - Teorico - Gestion de productos

Para que creamos un producto de sw?

- Para satisfacer al cliente
- p/tener usuarios logueados
- obtener dinero
- desarrollar una vision: cambiar el mundo

Evolución del producto de software(ampliar)

La evolución del producto de software se refiere a los cambios y mejoras que se realizan en el software después de su lanzamiento inicial. Esto puede incluir la corrección de errores, la actualización de funciones, la adición de nuevas funciones y la eliminación de funciones obsoletas. La evolución del producto de software es una parte importante del ciclo de vida del software, ya que permite que el software se adapte y evolucione con las necesidades cambiantes de los usuarios y del mercado. Además, la evolución del producto de software también puede ayudar a mantener la seguridad y la estabilidad del software a lo largo del tiempo.

La evolución del producto de software es un proceso continuo que requiere planificación y gestión adecuadas. Los cambios y mejoras deben ser cuidadosamente evaluados para garantizar que se ajusten a las necesidades de los usuarios y del mercado, y que no afecten negativamente la integridad y la estabilidad del software.

MVP - Minimum Viable Product

Conjunto de características mínimas que dan como resultado un producto de sw.

El primer paso no es crear el producto completo de SW, sino crear una minima funcional para validar si es atractiva para el cliente.

Sirve para verificar una hipótesis.

Sucede a veces , que una característica de nuestro MVP es la que más atención llama y nos permite validar la hipótesis que habíamos planteado. Eso nos lleva al siguiente concepto:

MMF: Minimum Marketable Feature, el MMF aporta valor. Supone una alta certeza sobre la utilidad y atractivo de dicha feature. **MVF**: Minimum Viable Feature. Característica minima que aun puede ser util para uso real y aprendizaje de los usuarios reales.

Si la MVF resulta exitosa, puede desarrollar más MMF.

MVP → MMP(con MMFs) → MMR (Minimum Marketable Release)

Valor vs Desperdicio: Lean Thinking, Lean Start-Up

- **Pensamiento Lean:** [Pensamiento Lean](#) es una metodología empresarial que tiene como objetivo promover la mejora continua y la creación de valor mientras se minimiza el desperdicio.

Build-Experiment-Learn Feedback loop. Permite descubrir las necesidades del cliente.

21/04 - Teórico - SCM

Software Configuration Management

Mención del SW en contexto: 4P

Personas, Producto - Acá está el Software, Proyecto, Proceso

Al pensar en software tenemos un conjunto de programas, procedimientos, reglas, etc.

El obj del prod de sw es brindar info, se necesita que los datos esten estructurados de determinada manera para poder procesarlo, se almacenan los datos de distintas maneras.

Evolucion del sw, la SCM ayuda que los cambios en el sw puedan ser gestionados de forma ordenada y **manteniendo la integridad del producto de sw**. Por ejemplo, conflicto, falta de concordancia etc. En este sentido, la gestión de la configuración de software (SCM) es una disciplina importante que ayuda a controlar la evolución del software. SCM se centra en identificar la configuración de los sistemas de software en diversos puntos en el tiempo y controlar sistemáticamente los cambios en la configuración durante todo el proceso de desarrollo de software.

- Cambios en el SW, y sus orígenes.

SCM es una disciplina de soporte, transversal al proyecto, relevante. No es actividad directa de ingeniería, sino que sirve como contención, para construir el prod de sw que queremos en contexto de disciplinas de soporte, como testing, calidad de prod y sw.

- Administración de configuración de software
- Aseguramiento de calidad:
 - Control de calidad de proceso
 - Control de calidad de producto
 - prueba del software

Definición de ANSI IEEE: Actividades fundamentales del SCM

- identificación de config,
- control de cambio de ítem de config,
- auditoría,

- reporte de control de estado.

Mantener la integridad del prod.

- Satisfacer los RF y RNF, asegura tener trazabilidad entre los req, y prod de sw construido.
- Problemas en el manejo de componentes, apaleados por las actividades fundamentales.

—seguir desde acá

Item de configuración

todos y cada uno de los artefactos que forman parte del producto o del proyecto, que pueden sufrir cambios o necesitan ser compartidos entre los miembros del equipo y sobre los cuales es necesario conocer su estado y evolución. SO, calendarización, documentación, manual de usuario, requerimientos. no se incluyen cuestiones de gestion de proyecto. Son artefactos de SW, un mail con requerimientos y me interese trazabilidad del mismo, doc de plan de proyecto, US. Condicion necesaria y suficiente (ampliar). Conf de sw: Conjunto de item de configs que lo conforman, y en la versión dada, es decir, la version del item de configuracion.

Items de config:

- Producto: ERS, Arquitectura,Codigo, Manual de usuario
- Proyecto: Plan de proyecto, Cronograma
- Iteracion: Plan de iteracion, reporte de defectos

Version

Estado de un item de configuracion en un momento de tiempo dado. Tiene determinada evolución, y cada uno de esos estados es una versión.

Variante: Mismo item de config que evoluciona por separado. puede ser porquer que el prod adopte distintas configs. una variante de android y otra de ios.

Config de SW: Conjunto de items con su correspondiente version en un momento determinado.

Repositorio: Lugar donde se almacena los items de configs. SVN, GIT.

Linea Base: Es una configuracion especial, revisada formalmente, sobre la que se llega a un acuerdo. Es base de desarrollos posteriores, se cambia a traves de un procedimiento formal. Permiten ir atras en el tiempo y reproducir el entorno de desarrollo dado del proyecto. Se actualiza segun criterio del equipo.

Creacion de Ramas:

Existe una principal, sirven para pruebas. Bifurcación de (rama, plan), hechas las pruebas se integra a la rama principal.

Identif de item de configuración

- Identifica cada item de config
- Convenciones y reglas de nombrado
- Definicion de Estructura de repositorio
- Ubicacion dentro de la estructura del repositorio

*Foto de items de config para proy de desarrollo de SW

Control de Cambios

- Origen en req de cambio, sobre item de linea base.
- Procedimiento formal que involucra diferentes actores (comité)
- Cuenta con un comité de control de cambios.

Auditoria de Gestion de Configuracion

Auditoria Funcional (FCA): Valida que el producto que estamos construyendo sea consistente con los req identificados. Matriz de rastreabilidad, rastrea determinado req con los modelos que se implementa, hasta el codigo y caso de prueba, se hace en distintos momentos.

Proceso: Validación: Construir el producto correcto.

Auditoria Fisica (PCA): Verifica que todo lo que se define en el **plan de configuracion** sea consistente con lo que estamos haciendo.

Proceso: Verificación: Construir el producto correctamente.

Reporte de Control del Estado

Responde a:

Estado del item?

Requerimientos de cambios aprobados y rechazados?

Diferencias entre 2 versiones dadas?

Plan de Gestion de Config

- Reglas de nombrado
- Procesos de Auditoria
- Plantillas que usamos
- Procedimiento formal de cambios
- Herramientas a utilizar para SCM.

Integracion entrega y despliegue continuos

Utilizar distintas herramientas para automatizar procesos de desarrollo de SW. *Foto del cel.

Integración Continua: Contruccion → Pruebas unitarias → Despliegue → Pruebas de aceptacion

Entrega continua: Idem arriba → (unica operación manual)Despliegue a Produccion

- Todas las otras “flechitas”, son procesos automatizados

Despliegue continuo: Idem entrega continua, pero el despliegue a producción es automatizado

SCM en Agile

Foto.

Teorico 5/5 - SCRUM

- SCRUM es un marco de trabajo, no una metodología agil.
- Principios del manifiesto agil.
- Se prefiere una cosa por sobre la otra en SCRUM pero debemos seguir haciendo la segunda, ej: software funcionando por encima de la documentación, pero se debe documentar.

Teorico 9/5 - SCRUM 2020

Anexo conceptos

- **Gestión de configuración de software:** La [gestión de configuración de software \(SCM\)](#) es una disciplina para controlar la evolución de los sistemas de software. SCM se centra en identificar la configuración de los sistemas de software en diversos puntos en el tiempo y controlar sistemáticamente los cambios en la configuración durante todo el proceso de desarrollo de software.
- **Producto mínimo viable:** [Producto mínimo viable \(MVP\)](#) es una técnica de desarrollo en la que se construye un nuevo producto o sitio web con suficientes características para satisfacer a los primeros adoptantes. El conjunto final completo de características solo se diseña y desarrolla después de considerar los comentarios de los usuarios iniciales del producto.
- **Wideband Delphi:** [Wideband Delphi](#) es una técnica de estimación para determinar el esfuerzo necesario para completar un proyecto. Se basa en el método Delphi, que implica que un grupo de expertos realicen estimaciones a través de rondas de discusión y refinamiento.
- **Puntos de historia:** [Puntos de historia](#) son una unidad de medida para estimar el esfuerzo general requerido para implementar completamente una característica de

un producto o una historia de usuario. Son una medida relativa que tiene en cuenta factores como la complejidad, el esfuerzo y la incertidumbre.

- **Línea base:** Una [línea base](#) es una versión aprobada formalmente de un elemento de configuración o producto. Actúa como punto de referencia, proporcionando una instantánea clara del estado del elemento o producto en un momento dado.
- **Scrum:** [Scrum](#) es un marco de trabajo ágil para la gestión de proyectos. Se enfatiza la colaboración, la retroalimentación y el desarrollo iterativo para ofrecer software funcional que cumpla con las necesidades del cliente y los objetivos empresariales.

Los principios del Manifiesto Ágil son los siguientes:

1. Individuos e interacciones sobre procesos y herramientas: Se valora más la colaboración directa entre las personas involucradas en el proyecto que los procesos o herramientas utilizados. Se reconoce que las interacciones efectivas y la comunicación directa son fundamentales para el éxito del proyecto.
2. Software funcionando sobre documentación extensiva: Se da prioridad a la entrega de software funcionando en lugar de enfocarse excesivamente en la documentación detallada. Se valora la capacidad de entregar valor tangible y funcionalidad a los clientes de manera rápida y constante.
3. Colaboración con el cliente sobre negociación de contratos: Se enfatiza la importancia de involucrar activamente a los clientes y obtener su colaboración y retroalimentación durante todo el proceso de desarrollo. Se valora la capacidad de adaptarse a las necesidades cambiantes del cliente y priorizar su satisfacción.
4. Respuesta ante el cambio sobre seguir un plan: Se reconoce que los requisitos y circunstancias pueden cambiar a lo largo del proyecto. Se valora la capacidad de responder y adaptarse a los cambios de manera ágil y efectiva, en lugar de aferrarse a un plan inicial rígido.

Estos principios promueven un enfoque ágil en el desarrollo de software, que se centra en la adaptabilidad, la colaboración, la entrega continua de valor y la respuesta efectiva a los cambios. Estos principios han influido en gran medida en la forma en que se desarrollan los proyectos y han dado lugar a enfoques ágiles populares como Scrum, Kanban y XP (Extreme Programming)