

Herramientas para la Gestión de Proyectos y Gestión de La Configuración

Herramientas para la Gestión de Proyectos y Gestión de La Configuración

- Herramientas para la Gestión de Proyectos y Gestión de La configuración
 - Artefactos de la gestión de Proyectos
 - Gestión de la configuración y Control de Versiones
 - Herramientas sugeridas
 - Qué tener en cuenta a la hora de elegir
 - Ejercicio por grupos

Artefactos de la gestión de Proyectos

Una vez decidido el sistema para gestionar un proyecto, es necesario conocer los artefactos que nos van a ayudar a realizar la gestión del proyecto, y posteriormente evaluar el estado del mismo.

Una vez decididos los artefactos con los que queremos trabajar, es necesario disponer de las herramientas que nos permitan disponer de estos artefactos.

Artefactos de la gestión de Proyectos

Una práctica habitual general es disponer de una herramienta en la que se puedan incluir *tareas o historas de usuario*, y asignarlas a un miembro del equipo, que pasará a ser el responsable de realizar o hacer que se realice esa tarea.

Artefactos de la gestión de Proyectos

En *Scrum* estas tareas se organizan en objetivos tácticos o estratégicos denominados *épicas*. El conjunto de *épicas* y tareas que no se están realizando en un Sprint se denomina *backlog* y es responsabilidad del *Product Owner* priorizar ese *backlog*, no obstante es responsabilidad de todo el equipo que las historias estén debidamente actualizadas, con las descripciones oportunas, y sus estimaciones.

También es responsabilidad de los miembros del equipo llevar la contabilidad del tiempo dedicado a cada una de las *historias de usuario*

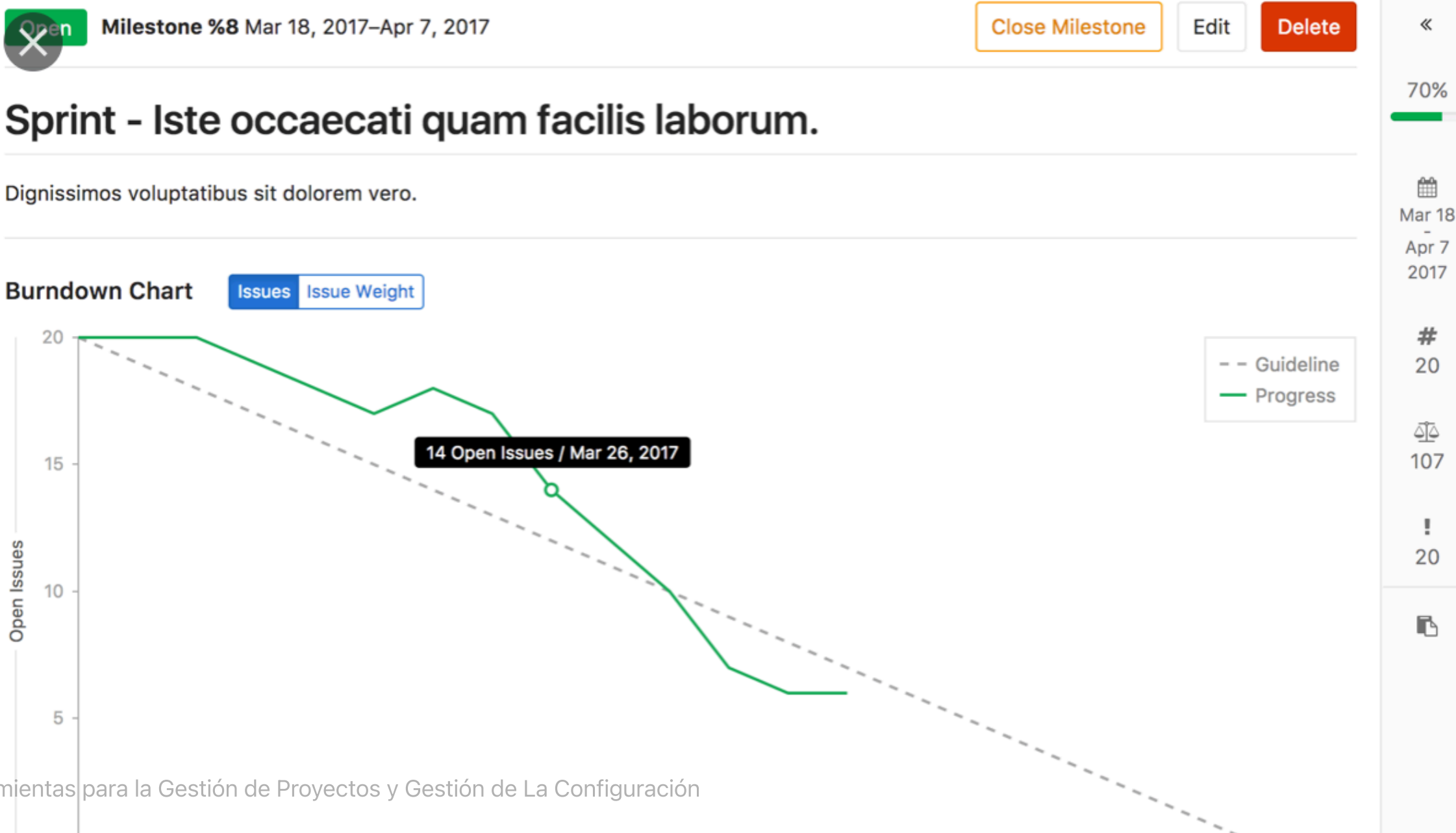
Artefactos de la gestión de Proyectos

En *Kanban* por ejemplo el backlog solo consiste en tareas que se priorizan, no obstante se pueden tener etiquetas dependiendo del ámbito.

Artefactos de la gestión de Proyectos

En *Scrum* una de las métricas más importantes es el *Burndown Chart* o gráfico de velocidad, que indica la cantidad de puntos de historia que es capaz de realizar un equipo por *Sprint*, y cómo esa velocidad cambia a lo largo del tiempo.

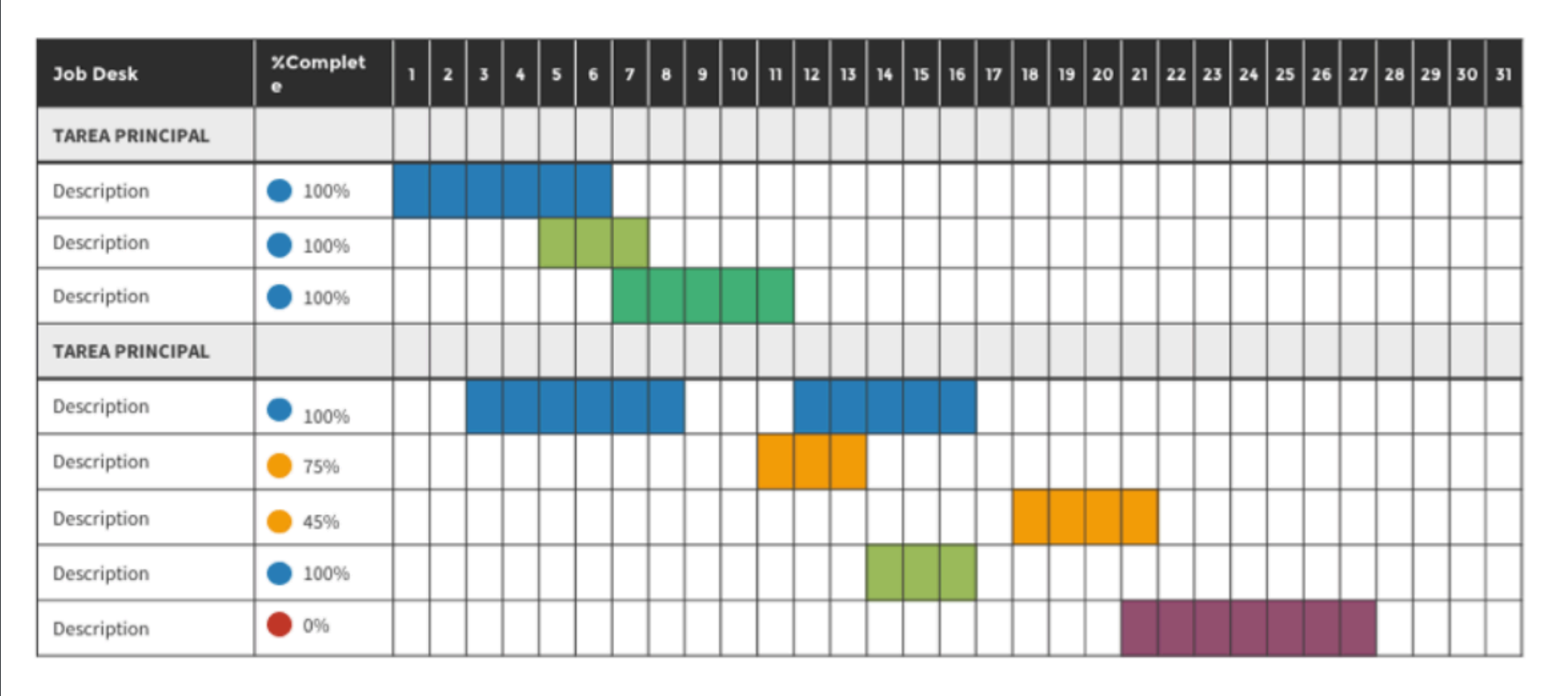
Artefactos de la gestión de Proyectos



Artefactos de la gestión de Proyectos

En waterfall los artefactos más habituales para conocer los estados de los proyectos son los diagramas de Gantt o Cronogramas.

Artefactos de la gestión de Proyectos



Artefactos de la gestión de Proyectos

Lo ideal es que la herramienta elegida sea común para toda una organización y se adapte a todas estas necesidades.

Gestión de la configuración y Control de Versiones

La gestión de la configuración es el conjunto de procesos destinados a asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un sistema de información, a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo.

Gestión de la configuración y Control de Versiones

Estos dos elementos, el control de cambios y control de versiones de todos los elementos del SI, facilitan también el mantenimiento de los sistemas al proporcionar una imagen detallada del sistema en cada etapa del desarrollo. La gestión de la configuración se realiza durante todas las fases del desarrollo de un sistema de información, incluyendo el mantenimiento y control de cambios, una vez realizada la puesta en producción.

Gestión de la configuración y Control de Versiones

Ejemplos de configuración de software:

- Ejecutables/ artefactos
- Código Fuente
- Modelos de datos
- Modelos de procesos
- Especificaciones de requisitos
- Pruebas

[De Wikipedia](#)

Herramientas sugeridas

- Github
- Gitlab
- Microsoft teams
- Jira
- Trello
- Gsuite
- Excel o similares
- ClickUp
- Asana
- OpenProject
- ProjectLibre
- Zentao

Focus: Notion como Espacio de Trabajo Unificado

A diferencia de herramientas verticales (solo gestión o solo documentos), Notion actúa como un **hub centralizado** que elimina los silos de información entre equipos técnicos y de producto.

- **Wiki de Ingeniería:** Centraliza guías de estilo, *onboarding* y documentación de infraestructura junto a las tareas, evitando la dispersión del conocimiento.
- **Gestión de Sprints:** Permite gestionar Épicas, Historias de Usuario y Bugs en la misma plataforma mediante bases de datos vinculadas y vistas personalizables (Tablero, Cronograma, Lista).

Notion: Integraciones y Automatización

Su valor diferencial reside en conectar la ejecución técnica con la documentación de negocio:

- **Sincronización con GitHub:** Vincula *Pull Requests* directamente a tareas de Notion. El estado de la tarea se actualiza automáticamente cuando el PR se fusiona, dando visibilidad a los PMs sin entrar al repositorio.
- **Sincronización con Jira:** Permite visualizar tickets de Jira dentro de Notion para equipos no técnicos.
- **Potenciado por IA:** Automatiza la redacción de historias de usuario a partir de especificaciones técnicas o resume reuniones de *daily*s.

Notion: Problema

El principal problema de Notion es el tiempo que se "pierde" en mantener todo en orden.

Si lo usamos como documentación hay que dedicar tiempo a mantenerlo integrado. Y si usamos IA para programar, los cambios tienden a ser tan rápidos que la documentación se queda obsoleta. Se pueden usar servidores MCP para mantener la documentación actualizada, pero requiere tiempo y esfuerzo.

Luego si lo utilizamos como herramienta de gestión de proyectos, hay que dedicar un tiempo a montar los automatismos y flujos de trabajo. No es solo crear tareas como en otras plataformas, es utilizar sus bases de datos y automatizaciones para sacarle el máximo partido.

Plan B: GitHub Projects + Wiki (El Ecosistema Nativo)

Si buscamos reducir la fricción del desarrollador o nuestra empresa no requiere que el conocimiento se comparta con, por ejemplo, financiero, la alternativa es vivir dentro de GitHub.

- **GitHub Projects:** Es la capa de gestión visual sobre el código.
 - **Automatización Nativa:** Al cerrar una *Issue* mediante un commit, la tarjeta se mueve sola en el tablero Kanban. Sin webhooks ni configuraciones complejas.
- **Documentación (Wiki vs. Docs/):**
 - **GitHub Wiki:** Útil para documentación de alto nivel (Onboarding, normas).
 - **Carpeta `/docs`:** Para documentación técnica viva. Al estar junto al código, evoluciona con él en el mismo *commit* ("Docs-as-code"), evitando la obsolescencia.

Comparativa: Fricción vs. Visibilidad

- Básicamente se basa en elegir entre:

◦ **Notion:** Mayor visibilidad y colaboración interdepartamental, pero con mayor fricción para desarrolladores

Qué tener en cuenta a la hora de elegir

- Precio
- Disponibilidad
- Esfuerzo de instalación/registro
- Curva de aprendizaje
- Funcionalidades que incluyen
- SaaS VS On-premise

Ejercicio por grupos

Pensar en los artefactos necesarios para la gestión del proyecto. Este ejercicio está incluido en la primera entrega. [RFI I](#). Es necesario documenta la decisión en un [ADR](#) o similar.

Decidid las herramientas necesarias para la gestión del proyecto. Este ejercicio está incluido en la primera entrega. [RFI I](#). Es necesario documenta la decisión en un [ADR](#) o similar.

Cada equipo deberá demostrar que ha utilizado la herramienta elegida a lo largo del desarrollo del proyecto.