

LA ETAPA DE LA DOCUMENTACIÓN

Documentación: Es la realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema. Las tareas de esta etapa se inician ya en el primera fase pero sólo finalizan una vez terminadas las pruebas, es importante porque permite comunicar detalles que al inspeccionar un sistema no se pueden alcanzar a comprender con facilidad, en dicha documentación se podría incluir (donde fuera necesario) el código fuente, los códigos de error y sus significados incluyendo la modelización (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc.

Todo ello con el fin de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones del sistema, sin ella, no se puede dar mantenimiento al software, los usuarios no pueden entrenar y prácticamente no pueden utilizar el software. Puede existir en formato electrónico, como archivo PDF o CD, o como libro.

Hay que tener en cuenta que en un proceso se pueden identificar 6 partes claves, que deben estar muy bien alineadas para que se cumpla con el objetivo de la documentación.

Partes del proceso

1. El objetivo del proceso.
2. Las entradas del proceso.
3. Las actividades.
4. El producto del proceso o resultado final.
5. Los indicadores de gestión.
6. Los controles del proceso.

Para un sistema de calidad son muy importantes estas partes porque aquí se sustenta la eficacia y la mejora continua de los procesos.

- El levantamiento del proceso se debe realizar con el personal a cargo del mismo.
- Empiece a describirlo en una “hoja en blanco”. No use por el momento formatos o plantillas pre-diseñadas. Déjelas para el final.
- El éxito de un proceso documentado, es la síntesis de las actividades que se realizan.

Normas para la documentación del software

El Instituto de Ingeniería Eléctrica y Electrónica (IEEE, por sus siglas en inglés) ha diseñado normas para la documentación de la industria del software. Se le denomina IEEE 1063-2001, "Estándar para la documentación de usuario de software", y detalla el contenido, la arquitectura y la importancia de un manual de documentación del software. La Organización Internacional de Normalización y la Comisión Electrotécnica Internacional (ISO e IEC, ídem) también han establecido normas para la documentación

del software. Sus estándares se llaman ISO/IEC 18019-2004 e ISO/IEC TR 9294. Se ocupan de la documentación y la gestión del software, respectivamente. Cada una de estas normas hace hincapié en que la documentación es una parte integral del ciclo de vida del software.

Documentación de software para desarrolladores

La importancia de la documentación para desarrolladores radica en el hecho de que la primera contiene información sobre las operaciones del sistema de software. Esta información posibilita la reproducción del software o su adaptación para mantenimiento. El tipo de información documentada suele valer muchísimo para la compañía de software. Dicha información podría perderse si no se documentan, ya que los desarrolladores van y vienen.

Documentación de software para usuarios finales

Los manuales de usuario final facilitan la interacción de dicho usuario con la computadora. Un manual de capacitación es un ejemplo de un manual orientado al usuario. Reduce el tiempo de aprendizaje para el uso productivo del software, ahorrando al usuario final muchísimo dinero. Los documentos del usuario final también pueden ser un manual de referencia, que detalle el funcionamiento del sistema, o una guía para administradores de sistema, que detalle la forma de operar y dar mantenimiento al sistema.

Consejos para mejorar la documentación.

- Mantenga el documento **simple y conciso**. Si bien debería ser técnicamente preciso, debería ser fácil de seguir.
- **Asegúrese de revisarlos al menos una vez al año**. O asigne un propietario de proceso que pueda hacer revisiones periódicas y notificar a otros sobre los cambios.
- Mantenga la **documentación separada para cada proceso** diferente para evitar confusiones.
- Cuando documente los procesos por primera vez, **evite cubrir toda la organización a la vez**. Comience desde un proceso único dentro de un departamento o un proceso importante común a toda la organización.
- Almacene los documentos en una ubicación a la que pueda **acceder fácilmente** cualquier persona que lo busque.
- Tenga los **documentos almacenados en línea en una ubicación central** a través de una herramienta de documentación de procesos. Esto ayudará automáticamente con las dos mejores prácticas mencionadas anteriormente.
- **Use ejemplos, gráficos, códigos de colores, capturas de pantalla, múltiples plataformas, etc.**, según sea necesario.

- **Cree una guía de documentación de procesos**, a la que cualquier persona pueda referirse como una plantilla estándar para documentar un proceso.
- **Utilice el material documental existente**, registros, entrevistas, estudios de caso, diarios de campo del personal del proyecto y el conocimiento de los empleados para recopilar información para la documentación del proceso.

Beneficios de la documentación de procesos

- Ayuda a que todos estén en la misma página al proporcionar **una vista común de los pasos del proceso**. Una visión común es necesaria para la **consistencia**.
- Cuando existe una documentación clara del proceso los cambios en los procesos se pueden realizar de manera oportuna y directa.
- Ayuda a identificar las condiciones ideales para **la colaboración efectiva y el rendimiento**.
- Ayuda a **identificar cuellos de botella e ineficiencias en los procesos** (incluso en aquellos procedimientos “bien refinados”) que, de lo contrario, podrían pasar desapercibidos.
- Un proceso claramente documentado **ahorra tiempo y evita errores** al eliminar cualquier confusión y al proporcionar instrucciones claras.
- **Disminuye costos y gasto innecesario de recursos**.
- Ayuda a **mejorar la calidad general de los procesos** y la calidad del resultado.
- Ayuda a **aumentar la productividad de los empleados, así como la satisfacción**.
- Dado que se registra claramente cómo deben llevarse a cabo los procesos, **facilita la externalización del trabajo**.
- La documentación de procesos también ayuda a **automatizar fácilmente los procesos** cuando sea necesario.

HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE.

1. Slack:

Esta herramienta es fantástica para mejorar la comunicación entre los equipos. La gran ventaja es la cantidad de integraciones que tiene de forma nativa.

Slack te permite organizar tus proyectos por canales y activar notificaciones de actualizaciones de nuevas implementaciones, alertas de rendimiento, además de facilitar la comunicación por medio de chats.

Es el software ideal para unir todas las informaciones que necesitas, contactar a tu equipo y recibir una respuesta al instante, compartir noticias de la empresa, visualizar las tareas y enviar todo tipo de archivos y añadir comentarios, y también etiquetar a personas. **Slack** es un aliado del trabajo colaborativo y funciona en toda ocasión. Por ejemplo, cuando el equipo necesita comenzar un proyecto, contratar a un nuevo empleado, implementar código, revisar un contrato de venta, finalizar el presupuesto del próximo año, medir una prueba A / B, planificar la próxima apertura de la oficina y más.

Slack es una app genial que utilizan todos los miembros del equipo de Apiumhub para comunicarse, ya que ayuda reduciendo la cantidad diaria de emails y además aumentó la comunicación entre los diferentes equipos.

También te ayuda a mantenerte concentrado ya que cada canal tiene su propio tema por lo que siempre sabes dónde encontrar información y dónde preguntar sobre un tema en concreto y recibir una respuesta al instante. Además dispones de una opción de chat que puedes utilizar para hablar directamente con cada miembro del equipo al instante. Por si esto no fuera suficiente, también ofrece una opción de arrastrar y soltar que va muy bien, una función que te permite compartir todo tipo de archivos y añadirles comentarios, o incluso etiquetar a gente. Muy recomendable.

2. Binfire.

Binfire es otra herramienta de software de gestión de proyectos online, en la que confían equipos tanto tradicionales como distribuidos. Ofrece métodos de colaboración en tiempo real, con funciones como una pizarra interactiva, un tablón de mensajes, cuadros, carpetas de proyectos, marcado de PDF colaborativo, notificaciones instantáneas, actualizaciones de estado y mucho más. En el gestor de tareas puedes encontrar herramientas de gestión de problemas, seguimiento de errores y secciones de colaboración de documentos.

Binfire tiene un conjunto de opciones realmente satisfactorio y las parejas que con una interfaz realmente intuitiva con un gran efecto. Ofrece todo lo que necesitas para planificar, rastrear y administrar sus proyectos remotos. La gestión de proyectos requiere planificación, unir a un equipo, crear tareas y un hito y asignarlos a los miembros del equipo.

3. UML

El Lenguaje Unificado de Modelado (**UML**), fue creado para forjar un lenguaje de modelado visual común y semántica y sintéticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. **UML** tiene aplicaciones más allá del desarrollo de software, por ejemplo en el flujo de procesos en la fabricación. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando diagramas UML. UML guarda una relación directa con el análisis y el diseño orientado a objetos.

Finalidad de UML según OMG.

El OMG define los propósitos de UML de la siguiente manera:

- ❖ Brindar a arquitectos de sistemas, ingenieros y desarrolladores de software, así como para el modelado de procesos de negocios y similares.
- ❖ Hacer progresar el estado de la industria permitiendo la interoperabilidad de herramientas de modelado visual de objetos. No obstante para habilitar un intercambio de modelos entre herramientas, se requiere de un acuerdo con respecto a la semántica y notación.

UML es popular entre los programadores, pero no suele ser usado por desarrolladores de base de datos. Una razón es sencillamente que los creadores de UML no se enfocaron en las bases de datos. A pesar de ello el UML es efectivo para el modelado de alto nivel de datos y se puede usar en diferentes tipos de diagramas UML.

UML cumple con los siguientes requerimientos:

- ✓ Establece una definición formal de un metamodelo común basado en el estándar MOF que especifique la sintaxis abstracta de UML.
- ✓ Brinda una explicación detallada de la semántica de cada concepto de modelado UML.
- ✓ Especifica los elementos de notación de lectura humana para representar los conceptos individuales de modelado UML, así como las reglas para combinarlos en una variedad de diferentes tipos de diagramas.

4. Jira

Jira es la **herramienta de desarrollo de software número uno entre los equipos ágiles**. Ayuda a planear, supervisar y gestionar proyectos de desarrollo de software. Además, ayuda a personalizar el flujo de trabajo, colaborar y lanzar buen software. Es de gran ayuda en la gestión de proyectos. Te permite gestionar el trabajo pendiente del equipo, visualizar el trabajo en curso y generar informes.

5. Trello

De hecho, es una de las aplicaciones de gestión de proyectos más utilizadas y mejor conocidas. La estructura de **Trello** está basada en la metodología kanban.

Todos los proyectos se representan con tableros que contienen listas. Cada lista tiene tarjetas de progreso que tú creas y que pueden moverse utilizando la función “arrastrar y soltar”. Los usuarios relacionados con cada tablero pueden asignar las tarjetas a otros usuarios. **Trello** tiene muchas funciones útiles, como, por ejemplo, escribir comentarios, añadir archivos adjuntos, establecer fechas de vencimiento, crear listados, añadir etiquetas de colores, integración con otras apps, etc.