

## **Anexo C. Instrumentos para medir la madurez basados en AMM.**

### **Instrumento 1**

Se tomo como base el modelo de madurez ágil o AMM por sus siglas en inglés, se toma este modelo porque está enfocando plenamente al desarrollo de software utilizando practicas agiles y de calidad para desarrollar el software, lo que permite que las empresas comiencen a conocer su madurez y puedan implementar técnicas directas dentro de sus proyectos de desarrollo de software y entregables.

### **Nivel 1**

#### **Ad Hoc (Inicial)**

(No se adapta en absoluto No hay objetivos de mejora del proceso definidos en este nivel no estructurado)

Las prácticas o el proceso de desarrollo de software son muy escasos en este nivel y no necesariamente repetibles. Las organizaciones no suelen proporcionar un entorno estable para el desarrollo. La empresa de nivel 1 no tiene definido un proceso ágil de desarrollo de software. Los principales problemas en este nivel están relacionados con los plazos excesivos, los retrasos, la comunicación, la calidad del software y el coste del desarrollo. Estas empresas funcionan a su manera y dependen de determinadas personas en lugar de todo el equipo. Si se describe el proceso de software tradicional, el éxito a este nivel depende de "la competencia y la heroicidad de las personas de la organización y no puede repetirse a menos que se asigne a las mismas personas al siguiente proyecto".

1. Al iniciar un proyecto de desarrollo nuevo:
  - a. Se cuenta con un proceso definido y es consistente. (Respuesta Correcta)
  - b. Depende de la persona que este asignada así se lleva el proyecto.
  - c. Se realiza una planificación detallada del proyecto antes de iniciar cualquier trabajo.
  - d. Se planifica de acuerdo con el valor que se identifica para el cliente.

2. ¿De los siguientes términos cuales se utilizan durante un proyecto?
  - a. Sprint, backlog, historias de usuario, tableros kanban.
  - b. Cronograma, actividades, gestión de adquisiciones.
  - c. Sprint, diagrama de Gantt, documento especificación de requisitos.
  - d. Ninguna de las anteriores.
3. ¿Para el seguimiento de proyectos utiliza alguna de las siguientes herramientas?
  - a. JIRA
  - b. Microsoft Project
  - c. Excel
  - d. Otra:
  - e. Ninguna
4. ¿Cuál es el objetivo principal del tablero kanban?
  - a. Almacenar y rastrear la información del equipo de desarrollo.
  - b. Mostrar el progreso del proyecto a los interesados.
  - c. Visualizar el flujo de trabajo y el estado de las tareas del proyecto.  
(Respuesta correcta)
  - d. Priorizar el trabajo en función de la importancia para el cliente.
5. ¿Cuál es el propósito de la reunión de planificación de sprint?
  - a. Revisar el trabajo completado durante el sprint anterior.
  - b. Establecer metas para el siguiente sprint.
  - c. Planificar el trabajo que el equipo llevará a cabo durante el próximo sprint.  
(Respuesta correcta)
  - d. Identificar los impedimentos que están afectando el progreso del equipo.

## **Nivel 2.**

El nivel 2 denota unas prácticas de desarrollo de software más estructuradas y completas que el nivel 1. Las organizaciones con capacidad de nivel 2 experimentaron menos problemas con su proceso de desarrollo de software que sus homólogas de nivel 1.

Los problemas de comunicación y las prácticas de codificación e integración siguen siendo un problema importante, junto con la retención de personal. Técnicamente, las dificultades para las empresas de nivel 2 se centraron en la comunicación (interacción mutua), las normas de codificación, los tiempos excesivos y la satisfacción del cliente.

### **Las siguientes preguntas tienen opciones de la siguiente manera**

5 Estoy muy de acuerdo

4 Estoy de acuerdo

3 Neutro

2 Estoy en desacuerdo

1 Estoy muy en desacuerdo

1. La planificación del proyecto en su organización está basada en el valor de negocio, utilizando herramientas y/o procesos definidos para su construcción
2. Se crean cronogramas de proyecto basado en la planificación
3. La estimación la realizan los desarrolladores.
4. Su organización gestiona los requerimientos del usuario con tarjetas de historia de usuario, con una estructura única y se cuenta con un plan de gestión de las historias de usuario.
5. El cliente está presente únicamente en la reunión de lanzamiento del proyecto.

6. Su empresa tiene un proceso para gestionar el cambio de requisitos.
7. Se identifican las incoherencias entre las tarjetas de historia y los requisitos originales del cliente.
8. Cada desarrollador realiza pruebas unitarias del código.

**Distribución de preguntar por área clave:**

Project Planning: 1. 2. 3.

Desarrollo por historias de usuario (ingeniería de requisitos): 4. 6. 7.

Disponibilidad del cliente en sitio: 5.

Introducción a TDD: 8.

### Nivel 3

El nivel 3 denota una mayor atención a las prácticas relacionadas con la gestión de las relaciones con los clientes, las entregas frecuentes, la programación en parejas, la comunicación, la codificación, las pruebas y la calidad del software.

#### **Las siguientes preguntas tienen opciones de la siguiente manera**

5 Estoy muy de acuerdo

4 Estoy de acuerdo

3 Neutro

2 Estoy en desacuerdo

1 Estoy muy en desacuerdo

1. ¿El representante del cliente y de la empresa están invitados a todas las sesiones de estimación del equipo?
2. ¿Se escriben historias de usuario en el proceso de gestión de proyectos?
3. ¿Se crean planes de proyecto como parte del proceso de gestión de proyectos?
4. ¿Las entregas frecuentes que crean un bucle de retroalimentación son parte del proceso de gestión de proyectos?
5. ¿El equipo de desarrollo y el cliente tienen interacción frecuente durante el proceso de gestión de proyectos?
6. ¿El código de producción se programa en parejas?
7. ¿Se fomenta y se implementan las mejores prácticas para la integración continua en el proceso de gestión de proyectos?

**Distribución de preguntar por área clave:**

Gestión de relación con el cliente: 1. 2. 3.

Entregas Frecuentes: 4.

Programación en pares: 5. 6.

Implementación e interacción: 7.

**Nivel 4**

Las empresas que se encuentran en este nivel de madurez están en condiciones de recopilar mediciones detalladas del proceso o las prácticas de desarrollo de software y de la calidad del producto; tanto las prácticas como los productos de desarrollo de software se comprenden y controlan cuantitativamente mediante mediciones detalladas.

El nivel mejorado del modelo AMM se centra en la gestión del proyecto, las horas de trabajo, la autoorganización del equipo, la evaluación de riesgos y está más relacionado con el equipo de desarrollo que con el producto en sí. Se trata de un atributo interno del equipo que no es directamente visible para el cliente. El nivel 4 denota un examen más activo y obligatorio del riesgo y respeto al equipo que va a desarrollar el sistema.

**Las siguientes preguntas tienen opciones de la siguiente manera**

5 Estoy muy de acuerdo

4 Estoy de acuerdo

3 Neutro

2 Estoy en desacuerdo

1 Estoy muy en desacuerdo

1. ¿Se utiliza una metodología definida para los proyectos?

2. ¿La principal métrica de seguimiento del progreso se basa en características, como historias completadas frente a historias restantes, gráficos de burn up o diagramas de flujo acumulativos?
3. ¿El equipo está facultado para tomar decisiones locales sobre cómo debe organizarse y llevarse a cabo el trabajo?
4. ¿El equipo del proyecto busca regularmente minimizar el despilfarro y generar eficiencias en sus procesos?
5. ¿Se reflexiona regularmente sobre el trabajo realizado y se ofrece un agradecimiento sincero a los colaboradores clave?
6. ¿Se prepara y se supervisa adecuadamente el riesgo del proyecto mediante pruebas de aceptación y mitigación de riesgos?
7. ¿Cree que el proceso de gestión del cambio que hay en el lugar tiende a fomentar la presentación de cambios que merezcan la pena?

**Distribución de preguntas por área clave:**

Gestión de proyectos: 1. 2.

Ritmo sostenible: 3.

Equipos autoorganizados: 4. 5. 7.

Evaluación de riesgos: 6.

## **Nivel 5**

Las empresas de este nivel mejoran continuamente sus procesos a través de la retroalimentación cuantitativa del proceso y la forma de probar ideas y tecnologías innovadoras. Las empresas que suben del nivel 4 al 5 deben disponer de una gran cantidad de datos métricos para gestionar el curso del proceso.

El nivel maduro de AMM aborda cuestiones de satisfacción del cliente y del desarrollador. Aquí decidimos tener en cuenta no sólo el proceso de software, sino también el resultado obtenido por el equipo.

### **Las siguientes preguntas tienen opciones de la siguiente manera**

- 5 Estoy muy de acuerdo
  - 4 Estoy de acuerdo
  - 3 Neutro
  - 2 Estoy en desacuerdo
  - 1 Estoy muy en desacuerdo
- 
1. ¿Todos los códigos pasan la prueba unitaria?
  2. ¿Se compensan las solicitudes de cambio y el trabajo de corrección de fallos con las características previstas para futuras iteraciones en colaboración con la transparencia del cliente?
  3. ¿Cada tarjeta de historia viene con una prueba de aceptación?
  4. ¿Se logra la satisfacción del cliente?
  5. ¿Se evalúan los efectos de realizar cambios en el proyecto?



6. ¿El cliente está disponible de manera regular durante el proyecto? (más de 2 horas diarias)

**Distribución de preguntas por área clave:**

Project planning: 1. 3. 4. 6.

Desarrollo de historias de usuario (Requerimientos): 2. 5.

### **Análisis de los resultados**

Número total de preguntas: 33

Número total de preguntas escala de Likert: 28

De acuerdo con los resultados obtenidos podemos concluir lo siguiente

KPA = Área clave de proceso.

- **Totalmente Conseguido:** Del 86% al 100% existen pruebas de un enfoque completo y sistemático y de la plena consecución de las prácticas clave definidas en la KPA evaluada. No existen debilidades significativas en la unidad organizativa definida.
- **Alcanzado en gran medida:** 51% a 85% hay evidencia de un enfoque sistemático sólido y un logro significativo de las prácticas clave definidas en la KPA evaluada. El rendimiento de las prácticas clave puede variar en algunas áreas.
- **Parcialmente alcanzado:** Del 16% al 50% hay pruebas de un enfoque sistemático sólido y de la consecución de las prácticas clave definidas en la KPA evaluada. Algunos aspectos de los logros pueden ser impredecibles.
- **No alcanzado:** Del 0% al 15% hay poca o ninguna evidencia del logro de las prácticas clave definidas en la KPA evaluada.

**Referencia:**

Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices Autores: Chetankumar Patel(1) and Muthu Ramachandran (2)

Se analizan los resultados por medio de la siguiente formula, para cada área de procesos, con su respectivo nivel.

*Puntos máx KPA: # Preguntas KPA \* 5*

$$\text{Resultado KPA} = \sum \text{Puntaje Preguntas KPA}$$

$$\text{Porcentaje} = \frac{\text{Resultado KPA}}{\text{Puntos máx KPA}}$$

De acuerdo con los resultados obtenidos podemos concluir lo siguiente

KPA = Área clave de proceso.

- Totalmente Conseguido: Del 86% al 100% existen pruebas de un enfoque completo y sistemático y de la plena consecución de las prácticas clave definidas en la KPA evaluada. No existen debilidades significativas en la unidad organizativa definida.
- Alcanzado en gran medida: 66% a 85% hay evidencia de un enfoque sistemático sólido y un logro significativo de las prácticas clave definidas en la KPA evaluada. El rendimiento de las prácticas clave puede variar en algunas áreas.
- Parcialmente alcanzado: Del 25% al 65% hay pruebas de un enfoque sistemático sólido y de la consecución de las prácticas clave definidas en la KPA evaluada. Algunos aspectos de los logros pueden ser impredecibles

- No alcanzado: Del 0% al 20% hay poca o ninguna evidencia del logro de las prácticas clave definidas en la KPA evaluada.

## Instrumento 2

Se tomo como base el modelo de madurez ágil o AMM por sus siglas en inglés, se toma este modelo porque está enfocando plenamente al desarrollo de software utilizando practicas agiles y de calidad para desarrollar el software, lo que permite que las empresas comiencen a conocer su madurez y puedan implementar técnicas directas dentro de sus proyectos de desarrollo de software y entregables. Se hace una traducción y se utiliza el instrumento propuesto por los autores Chetankumar Patel y Muthu Ramachandran.

## Nivel 1

### Ad Hoc (Inicial)

(No se adapta en absoluto No hay objetivos de mejora del proceso definidos en este nivel no estructurado)

Las prácticas o el proceso de desarrollo de software son muy escasos en este nivel y no necesariamente repetibles. Las organizaciones no suelen proporcionar un entorno estable para el desarrollo. La empresa de nivel 1 no tiene definido un proceso ágil de desarrollo de software. Los principales problemas en este nivel están relacionados con los plazos excesivos, los retrasos, la comunicación, la calidad del software y el coste del desarrollo. Estas empresas funcionan a su manera y dependen de determinadas personas en lugar de todo el equipo. Si se describe el proceso de software tradicional, el éxito a este nivel depende de "la competencia y la heroicidad de las personas de la organización y no puede repetirse a menos que se asigne a las mismas personas al siguiente proyecto".

6. Al iniciar un proyecto de desarrollo nuevo:
  - a. Se cuenta con un proceso definido y es consistente. (Respuesta Correcta)
  - b. Depende de la persona que este asignada así se lleva el proyecto.

- c. Se realiza una planificación detallada del proyecto antes de iniciar cualquier trabajo.
  - d. Se planifica de acuerdo con el valor que se identifica para el cliente.
7. ¿De los siguientes términos cuales se utilizan durante un proyecto?
- a. Sprint, backlog, historias de usuario, tableros kanban.
  - b. Cronograma, actividades, gestión de adquisiciones.
  - c. Sprint, diagrama de Gantt, documento especificación de requisitos.
  - d. Ninguna de las anteriores.
8. ¿Para el seguimiento de proyectos utiliza alguna de las siguientes herramientas?
- a. JIRA
  - b. Microsoft Project
  - c. Excel
  - d. Otra:
  - e. Ninguna
9. ¿Cuál es el objetivo principal del tablero kanban?
- a. Almacenar y rastrear la información del equipo de desarrollo.
  - b. Mostrar el progreso del proyecto a los interesados.
  - c. Visualizar el flujo de trabajo y el estado de las tareas del proyecto.  
(Respuesta correcta)
  - d. Priorizar el trabajo en función de la importancia para el cliente.
10. ¿Cuál es el propósito de la reunión de planificación de sprint?
- a. Revisar el trabajo completado durante el sprint anterior.
  - b. Establecer metas para el siguiente sprint.
  - c. Planificar el trabajo que el equipo llevará a cabo durante el próximo sprint.  
(Respuesta correcta)
  - d. Identificar los impedimentos que están afectando el progreso del equipo.

## Nivel 2.

El nivel 2 denota unas prácticas de desarrollo de software más estructuradas y completas que el nivel 1. Las organizaciones con capacidad de nivel 2 experimentaron menos problemas con su proceso de desarrollo de software que sus homólogas de nivel 1.

Los problemas de comunicación y las prácticas de codificación e integración siguen siendo un problema importante, junto con la retención de personal. Técnicamente, las dificultades para las empresas de nivel 2 se centraron en la comunicación (interacción mutua), las normas de codificación, los tiempos excesivos y la satisfacción del cliente.

### Las siguientes preguntas tienen opciones de la siguiente manera

- Si
- Parcialmente
- No

#### 1. Project Planning

- 1.1. ¿Se utiliza el juego de planificación para crear planes de proyecto?
- 1.2. ¿Se realiza una estimación del alcance del proyecto?
- 1.3. ¿Se utiliza la planificación de lanzamientos para crear calendarios?
- 1.4. ¿Se mide la velocidad del proyecto?
- 1.5. ¿Se divide el proyecto en iteraciones?
- 1.6. ¿Realizan los desarrolladores la estimación?
- 1.7. ¿Se utilizan técnicas de estimación pasada y de puntos funcionales para la estimación?
- 1.8. ¿Se basa la planificación del trabajo en el valor de negocio?
- 1.9. ¿El cliente o representante del negocio está presente o al menos invitado a todas las sesiones de estimación del equipo?
- 1.10. ¿El factor de iteración se basa en factores del proyecto como el tamaño, la complejidad y factores organizativos?
- 1.11. ¿Se logra conciliar el trabajo y el nivel de recursos?

#### 2. Desarrollo por historias de usuario (Ingeniería de requisitos)

- 2.1. ¿Se escriben historias de usuario para abordar la ingeniería de requisitos?
- 2.2. ¿Se define una estructura única de story cards?
- 2.3. ¿Existe un plan para gestionar las story cards?

- 2.4. ¿Se logra comprensión de las story cards?
- 2.5. ¿Se obtiene compromiso con las story cards?
- 2.6. ¿Se identifican incoherencias entre las story cards y los requisitos originales del cliente?
- 2.7. ¿Se obtienen las necesidades del cliente in situ?
- 2.8. ¿Existe un plan para el cambio de las story cards?
- 2.9. ¿Se gestiona el cambio de requisitos?
- 2.10. ¿Las story cards están escritas desde múltiples puntos de vista?
3. Disponibilidad del cliente en sitio
  - 3.1. ¿El cliente in situ es un experto en la materia?
  - 3.2. ¿Las decisiones de negocio son tomadas por el representante del cliente?
  - 3.3. ¿Se obtiene el compromiso de las tarjetas de historia del cliente in situ?
  - 3.4. ¿El cliente siempre está disponible?
4. Introducción a TDD
  - 4.1. ¿Se utilizan tarjetas CRC?
  - 4.2. ¿Las pruebas unitarias se identifican a partir de las tarjetas de tareas?

### Nivel 3

El nivel 3 denota una mayor atención a las prácticas relacionadas con la gestión de las relaciones con los clientes, las entregas frecuentes, la programación en parejas, la comunicación, la codificación, las pruebas y la calidad del software.

#### Las siguientes preguntas tienen opciones de la siguiente manera

- Si
- Parcialmente
- No

#### 1. Gestión de relación con el cliente

- 1.1. ¿Se define la metáfora del sistema, que permite al representante del cliente entender el sistema?
- 1.2. ¿El cliente y el representante del negocio están presentes o al menos invitados a todas las sesiones de estimación del equipo?
- 1.3. ¿Se fomenta la refactorización a través de un enfoque de "poco y a menudo" y se priorizan las refactorizaciones más grandes con el cliente?
- 1.4. ¿Se hacen pequeñas versiones frecuentes?
- 1.5. ¿Se escriben historias de usuario?
- 1.6. ¿Se crean planes de proyecto para crear un plan de proyecto?
- 1.7. ¿Se colabora eficazmente con el cliente?

#### 2. Entregas Frecuentes

- 2.1. ¿Se hacen pequeñas liberaciones frecuentes que crearán un bucle de retroalimentación?
- 2.2. ¿Sólo un par integra código a la vez?
- 2.3. ¿Se integra con frecuencia?

#### 3. Programación en pares

- 3.1. ¿Se desplaza a las personas?
- 3.2. ¿El cliente visita con frecuencia al equipo de desarrollo?
- 3.3. ¿Todo el código de producción se programa en parejas?
- 3.4. ¿Sólo una pareja integra código a la vez?
- 3.5. ¿Se utiliza la propiedad colectiva del código?



- 3.6. ¿Se recopilan y publican los datos de las horas extraordinarias?
4. Interacción mutua
    - 4.1. ¿Todo el código se programa en parejas?
    - 4.2. ¿Las story cards se escriben mediante la colaboración del cliente in situ y los desarrolladores?
    - 4.3. ¿Se comunica el resultado mediante pruebas de aceptación?
    - 4.4. ¿Se fomenta la refactorización a través de un enfoque de "poco y a menudo" y se repriorizan las refactorizaciones más grandes con el cliente?
    - 4.5. ¿Los miembros del equipo tienen un ambiente de trabajo abierto que apoya la colaboración y la conversación?
5. Test Driven Development (TDD)
    - 5.1. ¿Se codifica primero la prueba unitaria?
    - 5.2. ¿Se refactoriza siempre que sea posible?
    - 5.3. ¿Todo el código debe tener una prueba unitaria?
    - 5.4. ¿Todo el código debe pasar la prueba unitaria y la puntuación debe ser publicada antes de que pueda ser liberado?
    - 5.5. ¿Cuándo se encuentra un error se crean pruebas?
    - 5.6. ¿Se fomentan, recompensan e implementan en el proyecto las mejores prácticas de pruebas automatizadas?
    - 5.7. ¿Se realizan revisiones por pares?
    - 5.8. ¿Se analizan los resultados e identifican acciones correctivas?
6. Implementación e interacción
    - 6.1. ¿No se añade funcionalidad antes de tiempo?
    - 6.2. ¿Se integra a menudo?
    - 6.3. ¿Se utilizan pruebas automatizadas para apoyar las pruebas de integración frecuentes?
    - 6.4. ¿No se diseña por adelantado?
    - 6.5. ¿El equipo entrega contenido útil para revisión empresarial cada 1-4 semanas?
    - 6.6. ¿La lista de historias de usuario se vuelve a priorizar basándose en una evaluación actualizada del proyecto en cada límite de iteración?
    - 6.7. ¿Se alientan, recompensan e implementan las mejores prácticas para la integración continua?
    - 6.8. ¿Se prepara la integración del producto?
    - 6.9. ¿Se determina la secuencia de integración?

## 7. Estándares de codificación

- 7.1. ¿Los códigos deben escribirse según los estándares acordados?
- 7.2. ¿Se codifica primero la prueba unitaria?
- 7.3. ¿Todo el código de producción se programa por pares?
- 7.4. ¿Sólo un par integra código a la vez?
- 7.5. ¿Se utiliza la propiedad colectiva del código?

## Nivel 4

Las empresas que se encuentran en este nivel de madurez están en condiciones de recopilar mediciones detalladas del proceso o las prácticas de desarrollo de software y de la calidad del producto; tanto las prácticas como los productos de desarrollo de software se comprenden y controlan cuantitativamente mediante mediciones detalladas.

El nivel mejorado del modelo AMM se centra en la gestión del proyecto, las horas de trabajo, la autoorganización del equipo, la evaluación de riesgos y está más relacionado con el equipo de desarrollo que con el producto en sí. Se trata de un atributo interno del equipo que no es directamente visible para el cliente. El nivel 4 denota un examen más activo y obligatorio del riesgo y respeto al equipo que va a desarrollar el sistema.

### Las siguientes preguntas tienen opciones de la siguiente manera

- Si
- Parcialmente
- No

#### 1. Gestión de proyectos

- 1.1. ¿Se obtienen compromisos del plan?
- 1.2. ¿La métrica principal para el seguimiento del progreso se basa en características, como historias completadas versus historias restantes, gráficos o diagramas de flujo acumulativos?
- 1.3. ¿Se utiliza el proceso definido por el proyecto?
- 1.4. ¿Se integran los planes?
- 1.5. ¿Se coordina y colabora con las partes interesadas o el cliente in situ?
- 1.6. ¿Se utiliza la visión compartida del proyecto?

#### 2. Ritmo sostenible

- 2.1. ¿No se realizan horas extraordinarias (40 horas semanales)?
- 2.2. ¿El equipo directivo promueve un ritmo sostenible?
- 2.3. ¿Las responsabilidades son aceptadas por el equipo durante la vida del proyecto?
- 2.4. ¿Se faculta a los individuos para que tomen decisiones locales sobre cómo debe organizarse y llevarse a cabo el trabajo?
- 2.5. ¿Los miembros del equipo disponen de un entorno de trabajo abierto que favorezca la colaboración y la conversación?

3. Equipos autoorganizados
  - 3.1. ¿El equipo del proyecto se plantea hacer lo más sencillo que pueda funcionar?
  - 3.2. ¿Considera que el proceso de gestión del cambio existente tiende a fomentar la presentación de cambios que merezcan la pena?
  - 3.3. ¿El equipo cuestiona regularmente sus propios procesos y los de la empresa con el fin de minimizar el despilfarro y generar eficiencias?
  - 3.4. ¿El equipo, junto con el cliente y el PM, toma la mayor parte de las decisiones del proyecto?
  - 3.5. ¿El equipo reflexiona regularmente sobre el trabajo realizado y ofrece un agradecimiento sincero a los colaboradores clave?
  
4. Evaluación de riesgos
  - 4.1. ¿Se prepara la gestión de riesgos?
  - 4.2. ¿Se crean soluciones de pico para reducir el riesgo?
  - 4.3. ¿Se supervisa el riesgo del proyecto mediante pruebas de aceptación?
  - 4.4. ¿Se mitigan los riesgos?
  - 4.5. ¿Se identifican y analizan los riesgos?
  
5. Planeación de optimización de código
  - 5.1. ¿Se deja la optimización para el final?

## Nivel 5

Las empresas de este nivel mejoran continuamente sus procesos a través de la retroalimentación cuantitativa del proceso y la forma de probar ideas y tecnologías innovadoras. Las empresas que suben del nivel 4 al 5 deben disponer de una gran cantidad de datos métricos para gestionar el curso del proceso.

El nivel maduro de AMM aborda cuestiones de satisfacción del cliente y del desarrollador. Aquí decidimos tener en cuenta no sólo el proceso de software, sino también el resultado obtenido por el equipo.

### Las siguientes preguntas tienen opciones de la siguiente manera

- Si
- Parcialmente
- No

#### 1. Rendimiento del Proyecto

- 1.1. ¿Existe disponibilidad regular del cliente (más de 2 horas diarias)?
- 1.2. ¿No se realizan horas extraordinarias?
- 1.3. ¿Se logra la satisfacción del cliente?
- 1.4. ¿Todo el código debe pasar la prueba unitaria?
- 1.5. ¿Cada tarjeta de historia viene con las pruebas de aceptación?

#### 2. Prevención de defectos

- 2.1. ¿Se compensan las solicitudes de cambio y el trabajo de corrección de defectos con las características planificadas para futuras iteraciones en colaboración con el cliente?
- 2.2. ¿Se implementan las pruebas de aceptación con las story cards?
- 2.3. ¿Se determinan las causas de los defectos?
- 2.4. ¿Se seleccionan datos de defectos para análisis?
- 2.5. ¿Se abordan las causas de los defectos?
- 2.6. ¿Se implementa la propuesta de acción?
- 2.7. ¿Se evalúa el efecto de los cambios?

### **Análisis de los resultados**

Número total de preguntas: 109.

Número total de preguntas Si-No-Parcialmente: 104.

Se analizan los resultados por medio de la siguiente formula, para cada área de procesos, con su respectivo nivel.

$$\frac{\sum Si_n + \frac{1}{2} \sum P_n}{\sum T_n} * 100\%$$

*Si<sub>n</sub>: Número de respuestas SI*  
*P<sub>n</sub>: Número de respuestas Parciales*  
*T<sub>n</sub>: Número total de respuestas*

De acuerdo con los resultados obtenidos podemos concluir lo siguiente

KPA = Área clave de proceso.

- **Totalmente Conseguido:** Del 86% al 100% existen pruebas de un enfoque completo y sistemático y de la plena consecución de las prácticas clave definidas en la KPA evaluada. No existen debilidades significativas en la unidad organizativa definida.
- **Alcanzado en gran medida:** 51% a 85% hay evidencia de un enfoque sistemático sólido y un logro significativo de las prácticas clave definidas en la KPA evaluada. El rendimiento de las prácticas clave puede variar en algunas áreas.
- **Parcialmente alcanzado:** Del 16% al 50% hay pruebas de un enfoque sistemático sólido y de la consecución de las prácticas clave definidas en la KPA evaluada. Algunos aspectos de los logros pueden ser impredecibles.
- **No alcanzado:** Del 0% al 15% hay poca o ninguna evidencia del logro de las prácticas clave definidas en la KPA evaluada.

**Referencia:**

Agile Maturity Model (AMM): A Software Process Improvement framework for  
Agile Software Development Practices Autores: Chetankumar Patel(1) and Muthu  
Ramachandran (2)