

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DISEÑO DE UN MODELO DE GESTIÓN DE CALIDAD PARA DESARROLLO DE SOFTWARE EN LA UNIDAD DE ARTES GRÁFICAS DEL INSTITUTO GEOGRÁFICO MILITAR.

TESIS PREVIA A LA OBTENCIÓN DEL GRADO DE MÁSTER (MSc) EN
GESTIÓN DE LAS COMUNICACIONES Y TECNOLOGÍAS DE LA
INFORMACIÓN

CARRERA VILLACRÉS JOSÉ LUIS
joseluis.carrera@epn.edu.ec

JÁTIVA CHÁVEZ LIDIA DE LAS MERCEDES
mercedes.jativa@mail.igm.gob.ec

DIRECTOR: ING. CARLOS E. MONTENEGRO ARMAS
carlos.montenegro@epn.edu.ec

Quito, Enero de 2012

DECLARACIÓN

Nosotros José Luis Carrera Villacrés y Mercedes Játiva Chávez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría, que no ha sido para ningún grado o calificación profesional, y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la Escuela Politécnica Nacional, según lo establecido en la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad constitucional vigente.

Ing. José Luis Carrera V.

Ing. Mercedes Játiva Chavéz

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por José Luis Carrera Villacrés y Lidia de las Mercedes Játiva Chávez bajo mi supervisión.

Ing. Carlos Montenegro
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

A mi familia mi ejemplo de trabajo y superación continua.

A la Escuela Politécnica Nacional, mi alma máter, por darme la oportunidad de continuar por la senda del siempre anhelado desarrollo profesional.

José Luis Carrera V.

DEDICATORIA

A mis padres y hermanos, quienes siempre me han apoyado y con quienes sé que siempre contaré en todo momento, buenos y malos.

Familia es familia.

José Luis Carrera V.

AGRADECIMIENTOS

A Dios por darme la salud y todas las condiciones necesarias para culminar este proyecto.

A mi Padre Silvio por inculcarme desde mi infancia el amor por el estudio y la superación personal.

A mi Madre Lidia por estar siempre conmigo en todos mis pasos y ser mi gran amiga.

A mi esposo Juan Diego por su apoyo y amor incondicional.

Un agradecimiento especial al Director de Tesis Ing. Carlos Montenegro por su oportuna orientación y guía.

A mi compañero de Tesis José Luis Carrera con quien trabajé de manera eficiente y motivada con el fin de cumplir un mismo objetivo.

A esta Institución por ser una gran fuente de conocimiento y al Instituto Geográfico Militar por convertirse en un laboratorio de nuestros experimentos.

*Mil Gracias
Mercedes*

DEDICATORIA

Dedico este trabajo a mi hijo Ariel Eduardo la luz de mi vida y fuente de mi inspiración.

Mercedes

CONTENIDO

CAPÍTULO 1 SELECCIÓN DE METODOLOGÍA DE DESARROLLO Y MODELO DE CALIDAD	1
1.1. CARACTERIZACIÓN DE LOS PROYECTOS TÉCNICOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM	1
1.1.1. COMPARACIÓN DE METODOLOGIAS DE DESARROLLO VERSUS CARACTERÍSTICAS DE LOS PROYECTOS DE DESARROLLO INFORMÁTICO DE LA GESTION ARTES GRÁFICAS DEL IGM.	5
1.1.2. METODOLOGÍAS DE DESARROLLO ÁGIL	8
1.1.2.1. METODOLOGÍA DE DESARROLLO DYNAMIC SYSTEMS DEVELOPMENT METHOD(DSDM).....	10
1.1.2.2. MEDTODOLOGÍA DE DESARROLLO ÁGIL FEATURE DRIVEN DEVELOPMENT FDD.....	12
1.1.2.3. METODOLOGÍA DE DESARROLLO ÁGIL SCRUM.....	15
1.1.2.3.1. CARACTERÍSTICAS SCRUM.....	16
1.1.2.3.1.1. EMPIRISMO.....	16
1.1.2.3.1.2. PRIORIZACIÓN.....	16
1.1.2.3.1.3. TIMEBOXING.....	17
1.1.2.3.1.5. COLABORACIÓN Y COMUNICACIÓN ENTRE EL EQUIPO Y CLIENTE.....	17
1.1.2.3.1.6. FLEXIBILIDAD.....	18
1.1.2.3.1.7. REPLANIFICACIÓN AL INICIO DE CADA ITERACIÓN.....	18
1.1.2.3.1.8. ORGANIZACIÓN DE EQUIPOS Y ROLES.....	18
1.1.2.3.1.9. DUEÑO DEL PRODUCTO.....	19
1.1.2.3.1.10. FACILITADOR.....	19
1.1.2.3.1.11. EQUIPO	19
1.1.2.3.2. VENTAJAS SCRUM.....	19
1.1.2.3.2.1. CAMBIO ORGANIZACIONAL.....	20
1.1.2.3.2.2. ENTREGA RÁPIDA DE RESULTADOS.....	20
1.1.2.3.2.3. PRODUCTIVIDAD Y CALIDAD.....	20
1.1.2.3.2.4. MEJORA CONTINUA.....	21
1.1.2.3.2.5. COMUNICACIÓN DIARIA DEL EQUIPO Y COMPROMETIMIENTO.....	21

1.1.2.3.3. DESVENTAJAS SCRUM.....	21
1.1.2.4. METODOLOGÍA DE DESARROLLO ÁGIL EXTREME PROGRAMING XP.....	21
1.1.2.4.1. OBJETIVOS DE LA METODOLOGÍA XP.....	21
1.1.2.4.2. FASES DE XP.....	21
1.1.2.4.2.1. FASE DE PLANIFICACIÓN.....	24
1.1.2.4.2.1.1 REDACCIÓN DE HISTORIAS DE USUARIO.....	24
1.1.2.4.2.1.2 ELABORACIÓN DEL PLAN DE ENTREGAS.....	24
1.1.2.4.2.1.3 CONTROL DE LA VELOCIDAD DEL PROYECTO.....	24
1.1.2.4.2.1.4 DIVISIÓN DEL PROYECTO EN ITERACIONES.....	24
1.1.2.4.2.1.5 DEFINICIÓN DEL PLAN DE ITERACIÓN.....	24
1.1.2.4.2.1.6 ROTACIÓN DE PERSONAL.....	26
1.1.2.4.2.2. FASE DE DISEÑO.....	26
1.1.2.4.2.3. FASE DE DESARROLLO.....	26
1.1.2.3.2.4. FASE DE PRUEBAS.....	27
1.2. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO PARA PROYECTOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM. 28	
1.3. MODELOS DE CALIDAD PARA PROYECTOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM.....	29
1.3.1. ISO 9001:2008.....	32
1.3.2. CMMI.....	34
1.3.2.1 NIVELES CMMI.....	37
1.3.2.1.1 NIVEL 1.....	37
1.3.2.1.2 NIVEL 2.....	37
1.3.2.1.3 NIVEL 3.....	37
1.3.2.1.4 NIVEL 4.....	37
1.3.2.1.5 NIVEL 5.....	37
1.4. SELECCIÓN DEL MODELO DE CALIDAD PARA PROYECTOS INFORMÁTICOS DE LA GESTION ARTES GRÁFICAS DEL IGM.	38
CAPÍTULO 2 . CONSTRUCCIÓN DE UN MODELO DE GESTIÓN DE CALIDAD PARA DESARROLLO DE PROYECTOS INFORMÁTICOS DE GAG. IGM.	44

APLICABLES AL MODELO DE GESTIÓNPROPUESTO.....	47
2.1.1. KPAs NO APLICABLES NIVEL 2 CMMI.....	47
2.1.1.1. KPA4: ADMINISTRACIÓN Y CONTRATACIÓN DE PROVEEDORES.....	47
2.1.1.2. KPA5: MÉTRICAS Y ANÁLISIS.....	48
2.2. DETERMINACIÓN Y CLASIFICACIÓN DE KPAsNIVEL 3 CMMI APLICABLES AL MODELO DE GESTIÓN PROPUESTO.....	49
2.2.1. KPAs NO APLICABLES NIVEL 3 CMMI.....	50
2.2.1.1. KPA 9: SOLUCIÓN TÉCNICA.....	50
2.2.1.2. KPA17: ANÁLISIS DE DESICIONES Y RESOLUCIONES.....	50
2.2.1.3. KPA18: AMBIENTE DE DESARROLLO INTEGRADO PRODUCTO Y PROCESO.....	51
2.2.1.4. KPA 21:GESTIÓN INTEGRADA DE PROVEEDORES.....	51
2.3. MODELO DE GESTIÓNPROPUESTO MEDIANTE KPAs APLICABLESDE LOS NIVELES 2 Y 3 CMMI	52
2.3.1. ÁMBITO ORGANIZACIONAL.....	52
2.3.1.1 KPA13: ENFOQUE EN ELPROCESO DE LA ORGANIZACIÓN.....	52
2.3.1.2 KPA14: DEFINICIÓN DEL PROCESO ORGANIZACIONAL.....	53
2.3.1.3 KPA15: CAPACITACIÓN ORGANIZACIONAL.....	53
2.3.1.4 KPA22: INTEGRACIÓN DEL AMBIENTE ORGANIZACIONAL....	54
2.3.2. ÁMBITO PRODUCTO DE SOFTWARE	54
2.3.2.1 KPA1: GESTIÓN DE REQUERIMIENTOS.....	54
2.3.2.1.1 GESTIÓN DE LA PILA DE PRODUCTO.....	55
2.3.2.1.2 SEGUIMIENTO DE REQUERIMIENTOS EN LA PILA DE PRODUCTO.....	58
2.3.2.1.3 CONGELACIÓN DE REQUERIMIENTOS DURANTE CADA SPRINT.....	59
2.3.2.1.4 INTEGRACIÓN CONTÍNUA.....	60
2.3.2.1.5 RETROALIMENTCACIÓN O INTERACCIÓN CONTINUA CON EL CLIENTE.....	60
2.3.2.2 KPA2: PLANIFICACIÓN DE PROYECT DE SOFTWARE.....	61
2.3.2.2.1 PLANIFICACIÓN SCRUM-XP AL INICIO DE CADA PROYECTO.....	62
2.3.2.2.2 REUNIONES DE PLANIFICACIÓN DE SPRINT AL INICIO DE CADA UNO DE ESTOS.....	64
2.3.2.2.3 INCLUSIÓN DE HISTORIAS EN EL SPRINT.....	68
2.3.2.2.4 DEFINICIÓN DE LA DURACIÓN DEL SPRINT.....	70

2.3.2.2.5	DEFINICIÓN DEL SITIO Y LA HORA PARA EL SCRUM DIARIO.....	70
2.3.2.2.6	ELABORACIÓN DE HISTORIAS TÉCNICAS.....	71
2.3.2.3	KPA3: SEGUIMIENTO Y SUPERVISIÓN DE PROYECTOS DE SOFTWARE.....	72
2.3.2.3.1	SCRUM DIARIO.....	72
2.3.2.3.2	MEDICIÓN DEL AVANCE CON GRÁFICAS DE BACKLOG.....	74
2.3.2.4	KPA6: ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE.....	77
2.3.2.4.1	PRUEBAS DE FUNCIONALIDAD.....	80
2.3.2.4.2	REVISIÓN DE PRESENTACIÓN.....	81
2.3.2.4.3	REVISIÓN DE ESTANDARES DE CODIFICACIÓN.....	81
2.3.2.5	KPA7: GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE.....	82
2.3.2.5.1	ESTABLECER LÍNEAS BASE.....	85
2.3.2.5.2	ESTABLECER CONTROL DE CAMBIOS.....	87
2.3.2.5.3	ESTABLECER INTEGRIDAD.....	87
2.3.2.6	KPA8: DESARROLLO DE REQUERIMIENTOS.....	88
2.3.2.7	KPA10: INTEGRACIÓN DEL PRODUCTO.....	89
2.3.2.8	KPA11: VERIFICACIÓN.....	91
2.3.2.9	KPA12: VALIDACIÓN.....	95
2.3.2.10	KPA16: GESTIÓN DE RIESGOS.....	95
2.3.2.11	KPA19: GESTIÓN INTEGRADA DEL PROYECTO.....	97
2.3.2.12	KPA20: INTEGRACIÓN DEL EQUIPO O COORDINACIÓN INTERGRUPAL.....	99
CAPÍTULO 3 : APLICACIÓN A UN CASO DE ESTUDIO		102
3.1.	SELECCIÓN DEL CASO DE ESTUDIO.....	102
3.1.1.	DESCRIPCIÓN DEL CASO DE ESTUDIO SELECCIONADO....	102
3.2.	IMPLEMENTACIÓN DEL CASO DE ESTUDIO SELECCIONADO UTILIZANDO EL MODELO DE GESTIÓN DESARROLLADO.	104
3.2.1.	ÁMBITO ORGANIZACIONAL.....	104
3.2.1.1	KPA13: ENFOQUE EN EL PROCESO DE LA ORGANIZACIÓN...103	
3.2.1.2	KPA14: DEFINICIÓN DEL PROCESO ORGANIZACIONAL.....104	
3.2.1.2.1	MAPA DE PROCESOS DEL INSTITUTO GEOGRÁFICO MILITAR.....	105
3.2.1.2.1.1	ESTRUCTURA DE PROCESOS IGM.....	105
3.2.1.2.1.1.1	GESTIÓN TECNOLÓGICA.....	106
3.2.1.3	KPA15: CAPACITACIÓN ORGANIZACIONAL.....107	
3.2.1.4	KPA22: INTEGRACIÓN DEL AMBIENTE ORGANIZACIONAL...107	
3.2.1.4.1	METODOLOGÍA.....	108

3.2.1.4.2	MISIÓN.....	108
3.2.1.4.3	VISIÓN.....	108
3.2.1.4.4	OBJETIVOS EMPRESARIALES-ESTRATEGIAS.....	108
3.2.1.4.4.1	OBJ. ESTRATEGICOS Perspectiva Cliente.....	108
3.2.1.4.4.2	OBJ. ESTRATEGICOS Perspectiva Interna.....	109
3.2.1.4.4.3	OBJ. ESTRATEGICOS Perspectiva Desarrollo....	109
3.2.2.	ÁMBITO PRODUCTO DE SOFTWARE	111
3.2.2.1	KPA1: GESTIÓN DE REQUERIMIENTOS.....	110
3.2.2.1.1	GESTIÓN DE LA PILA DEL PRODUCTO.....	111
3.2.2.1.2	SEGUIMIENTO DE REQUERIMIENTOS DE LA PILA DEL PRODUCTO.....	117
3.2.2.1.3	CONGELACIÓN DE REQUERIMIENTOS DURANTE CADA SPRINT.....	118
3.2.2.1.4	INTEGRACIÓN CONTINUA.....	118
3.2.2.1.5	INTERACCIÓN CONTINUA CON EL CLIENTE.....	119
3.2.2.2	KPA2: PLANIFICACION DE PROYECTOS DE SOFTWARE.....	120
3.2.2.2.1	PLANIFICACIÓN SCRUM-XP AL INICIO DE CADA PROYECTO.....	120
3.2.2.2.2	PLAN. DE SPRINT AL INICIO DE CADA UNO.....	120
3.2.2.2.2.1	DEFINICION DE LA META DE SPRINT.....	121
3.2.2.2.2.2	DEFINICIÓN DE LA LISTA DE MIEMBROS DEL EQUIPO DE TRABAJO Y EL NIVEL DE DEDICACIÓN....	121
3.2.2.2.2.3	INCLUSIÓN DE HISTORIAS EN EL SPRINT....	123
3.2.2.2.2.4	DEF. DE LA DURACIÓN DEL SPRINT	124
3.2.2.2.2.5	DEF. DEL SITIO Y HORA PARA EL SCRUM DIARIO.....	124
3.2.2.2.2.6	ELABORACIÓN DE HISTORIAS TECNICAS...	124
3.2.2.3	KPA3: SEGUIMIEN TOY SUPERVISIÓN DE PROYECTOS DE SOFTWARE.....	127
3.2.2.3.1	SCRUM DIARIO No 1.....	127
3.2.2.3.2	MEDICIÓN DE AVANCE EN GRÁFICAS DE BACKLOG	131
3.2.2.4	KPA6: ASEGURAMIENTO DE LA CALIDAD DEL SW	134
3.2.2.4.1	PRUEBAS DE FUNCIONALIDAD.....	134
3.2.2.4.2	REVISIÓN DE PRESENTACIÓN.....	135
3.2.2.4.3	REVISIÓN DE ESTANDARES DE CODIFICACIÓN.....	136
3.2.2.6.	KPA8: DESARROLLO DE REQUERIMIENTOS.....	138
3.2.2.7	KPA10: INTEGRACIÓN DE PRODUCTO.....	139
3.2.2.8	KPA11: VERIFICACIÓN.....	139
3.2.2.9	KPA12: VALIDACIÓN.....	143
3.2.2.10	KPA16: GESTIÓN DE RIESGOS.....	145
3.2.2.11	KPA19: GESTIÓN INTEGRADA DEL PROYECTO.....	146
3.2.2.12	KPA20: INTEGRACIÓN DEL EQUIPO O COORDINACIÓN INTERGRUPAL.....	147

CAPÍTULO 4 CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES ..	150
4.1. CONCLUSIONES	150
4.2. RECOMENDACIONES	154
BIBLIOGRAFIA	156

LISTA DE TABLAS

Tabla 1.1 Comparación Met de Desarrollo vs Características Proy. IGM	7
Tabla 1.2 Comparación de metodologías ágiles	29
Tabla 2.1: KPAs Nivel 2 CMMI Repetible	45
Tabla 2.2: KPAs Nivel 3 CMMI Definido	46
Tabla 2.3: Selección y clasificación KPAs aplicables NIVEL 2 CMMI	47
Tabla 2.4: Selección y clasificación KPAs aplicables NIVEL 3 CMMI	50
Tabla 2.5: Definición de roles del proyecto	56
Tabla 2.6: Elemento de la pila del producto	58
Tabla 2.7: Elementos de la pila del producto presentación general	58
Tabla 2.8: Definición de la Meta de Sprint.....	67
Tabla 2.9: Definición del Equipo y Nivel de Dedicación	68
Tabla 2.10: División de Historias para el Sprint Ilustración 1	70
Tabla 2.11: Historia Técnica.....	71
Tabla 2.12: Registro de Scrum Diario	74
Tabla 2.13: Tabla de Sprint	75
Tabla 2.14: Revisión de Funcionalidad SPRITN	81
Tabla 2.15: Revisión de Presentación SPRITN.....	82
Tabla 2.16: Revisión Estándares de Codificación SPRITN	82
Tabla 2.17: Gestión de Configuración de Software	86
Tabla 2.18: Línea Base	87
Tabla 2.19: Nuevo Elemento de Pila de Producto.....	89
Tabla 2.20: Integración de Producto	90
Tabla 2.21: Sprint entregado.....	92
Tabla 2.22: Verificación	93
Tabla 2.23: Gestión de Riesgos	98
Tabla 3.1: Enfoque en Proceso de la Organización	105
Tabla 3.2: Definición de roles del proyecto	112
Tabla 3.3: Pila de Producto Visión General.....	113
Tabla 3.4: Elemento 001 Pila de Producto	114
Tabla 3.5: Elemento 002 Pila de Producto	114

Tabla 3.6: Elemento 003 Pila de Producto	115
Tabla 3.7: Elemento 004 Pila de Producto	116
Tabla 3.8: Elemento 005 Pila de Producto	116
Tabla 3.9: Elemento 006 Pila de Producto	117
Tabla 3.10: Elemento 007 Pila de Producto	118
Tabla 3.11: Participantes Planificación de Sprint	120
Tabla 3.12: Participantes Scrum diario.....	120
Tabla 3.13: Verificación para planificación de Sprint.....	121
Tabla 3.14: Meta de Sprint No 1.....	122
Tabla 3.15: Definición Equipo de Trabajo para Sprint No. 1	123
Tabla 3.16: División de Historias para Sprint No. 1	124
Tabla 3.17: Instalación de Base de Datos Sprint No. 1	126
Tabla 3.18: Instalación de Periféricos Sprint No. 1.....	126
Tabla 3.19: Instalación y Configuración clientes Sybase Sprint No. 1.....	127
Tabla 3.20: Scrum diario Sprint No. 1	131
Tabla 3.21: Backlog al final del Sprint No. 1.....	133
Tabla 3.22: Revisión de la Funcionalidad Sprint No. 1	136
Tabla 3.23: Revisión de Presentación Sprint No. 1	137
Tabla 3.24: Revisión de Estándares de Codificación Sprint No. 1	137
Tabla 3.25: Elemento de Gestión Librería para Manejo de Cámara Canon...	138
Tabla 3.26: Elemento de Gestión Librería para Manejo de Pad de firmas TOPAZ	139
Tabla 3.27: Elemento de Gestión Librería para Manejo de Lector Huellas	139
Tabla 3.28: Integración de Sprints.....	140
Tabla 3.29: Verificación Sprint 1 verificación 2.....	141
Tabla 3.30: Verificación Sprint 1 verificación 3.....	142
Tabla 3.31: Verificación Sprint 1 verificación 4.....	143
Tabla 3.32: Verificación Sprint 1 verificación 5.....	143
Tabla 3.33: Verificación Sprint 1 verificación 6.....	143
Tabla 3.34: Verificación Sprint 1 Iteración 2.....	144
Tabla 3.35: Validación 1 Sprint 1 Iteración 1	145
Tabla 3.36: Validación 2 Sprint 1 Iteración 1	145
Tabla 3.37: Validación 1 Sprint 1 Iteración 2.....	146

Tabla 3.38: Validación 1 Sprint 1 Iteración 3.....	146
Tabla 3.39: Gestión de Riesgos	147
Tabla 3.40: Coordinación Intergruppal	148

LISTA DE FIGURAS

Figura 1.1: Fases Metodología de desarrollo FDD.....	13
Figura 1.2:Elaboración del Plan de Iteración.....	25
Figura 1.3: Mejora Continua del Sistema de Gestión de Calidad.....	34
Figura 1.4: Estructura CMMI en un nivel de madurez	36
Figura 1.5: Niveles de Madurez CMMI	38
Figura 1.6: Arquitectura MoProSoft	40
Figura 1.7: Procesos MoProSoft	41
Figura 2.1: Variables del Elemento de la Pila de Producto	65
Figura 2.2: Gráfico de Backlog	76
Figura 2.3: Pruebas de software	79
Figura 3.1: Mapa de Procesos IGM.....	106
Figura 3.2: Avances Backlog Sexto día Sprint No. 1.....	134
Figura 3.3: Avances Backlog al Final del Sprint No. 1.....	134

RESUMEN

El presente trabajo plantea el Diseño de un Modelo de Gestión de Calidad para Desarrollo de Software en la Unidad de Artes Gráficas del Instituto Geográfico Militar.

Para lo cual se combina un modelo de calidad con una metodología de desarrollo para ser aplicado en un tipo de proyectos informáticos caracterizados principalmente por su corta duración requerimientos cambiantes y complejidad tecnológica.

En el Capítulo 1: Establece las características de los proyectos técnicos de la Gestión Artes Gráficas del Instituto Geográfico Militar, para establecer la metodología apropiada para el desarrollo de dichos proyectos. En este capítulo también analiza diferentes modelos de calidad para desarrollo de software, determinando el más apropiado a utilizar en función caracterización de los proyectos abordados en este trabajo.

Capítulo 2: Se combina KPAs de los niveles de madurez 2 y 3 de CMMI aplicables al tipo de proyectos caracterizados en el Capítulo 1, con prácticas de las metodologías de desarrollo ágil SCRUM-XP, obteniendo de esta forma el Modelo Calidad que es objetivo del trabajo.

Capítulo 3: En este capítulo se aplica el modelo propuesto a un caso de estudio real de la Gestión Artes Gráficas del Instituto Geográfico Militar, tomando el Proyecto para la Emisión de Tarjetas de Identificación para Las Fuerzas Armadas del Ecuador, DIRMOV.

Capítulo 4: Muestra las conclusiones y recomendaciones obtenidas como resultado del proyecto.

CAPÍTULO 1 SELECCIÓN DE METODOLOGÍA DE DESARROLLO Y MODELO DE CALIDAD

1.1. CARACTERIZACIÓN DE LOS PROYECTOS TÉCNICOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM

El Instituto Geográfico Militar del Ecuador es el organismo autorizado por el estado para la generación y regulación de la base de datos Cartográfica - Geográfica del país, además el IGM es una entidad estatal que posee la tecnología e infraestructura necesaria para suministrar soluciones gráficas y de seguridad documentaría tanto al mercado nacional como internacional; además el IGM cuenta con una extensión cultural en el campo la astronomía y ciencias afines que brinda sus servicios a la comunidad ecuatoriana.

La División Artes Gráficas debe realizar el diseño y la impresión de las especies valoradas y soluciones gráficas de seguridad en cumplimiento al Decreto No. 014 y en el cumplimiento de la misión del Instituto Geográfico Militar.

Es prioridad para el Instituto Geográfico Militar optimizar los sistemas informáticos técnicos que forman parte de soluciones gráficas de seguridad documentaria, de tal forma que le permita al IGM llevar a cabo sus metas y objetivos organizacionales relacionados con la cadena productiva de la División de Artes Gráficas.

Los proyectos técnicos informáticos deben estar acorde a las necesidades del usuario interno o externo. Estos sistemas de información deben contribuir a desarrollar nuevas soluciones, las cuales servirán para el mejoramiento de los productos y soluciones gráficas de seguridad documentaria de la división de Artes Gráficas del IGM.

La calidad en el proceso de desarrollo de software tiene cada vez mayor importancia dado a que con el tiempo los usuarios de sistemas informáticos se vuelven cada vez más exigentes y críticos a la hora de evaluar y usar un producto de software.

El usuario final optará por el software que le brinde confiabilidad y seguridad respondiendo a sus exigencias, es por esto que es de gran importancia el implementar estándares y prácticas que aseguren un nivel de calidad en los procesos de desarrollo de aplicaciones informáticas.

En la Unidad Artes Gráficas del IGM se generan proyectos en los cuales necesariamente se debe incluir el desarrollo de aplicaciones informáticas que por la tecnología a utilizar eleva el grado de complejidad, sin embargo en la parte informática de estos proyectos actualmente no se cuenta con ninguna metodología formal que asegure un desarrollo ágil combinando estándares de calidad en todas fases de desarrollo.

La constante evolución de la tecnología para generación de documentación de seguridad ha hecho que varios productos de artes gráficas adquieran una funcionalidad que debe ser complementada con la aplicación de tecnologías de información, estas tecnologías de información se encuentran presentes en el proceso de producción y en ciertos productos incluso es necesario implementarlas a nivel de usuario final.

CMM emplea una escala que va en rango desde al 5 para determinar el grado de madurez de una organización, y por tanto el grado de garantía que esta ofrece en la calidad de sus productos, predictibilidad en los resultados y eficiencia en el desarrollo de software.

Las empresas que no tienen implementado una metodología son consideradas en CMM en el nivel 1 de madurez. Esto no quiere decir que necesariamente que estas empresas van a producir software de mala calidad, de forma ineficiente y con retraso, sino que las probabilidades de cumplir las tres facetas es baja.

Los resultados en los proyectos en este tipo de empresas dependen de las personas que los desarrollan.

El éxito o fracaso de las organizaciones que trabajan sin metodologías depende del conocimiento tácito de su personal, teniendo en cuenta que se trata de conocimiento adquirido externamente o por esfuerzo propio.

En el IGM hasta el momento no se ha implementado ninguna metodología formal para gestión y desarrollo de proyectos informáticos, por esta razón y de acuerdo a CMMI, el IGM se encuentra en el nivel 1 de madurez.

Los proyectos técnicos informáticos que se generan en la División Artes Gráficas del IGM presentan las siguientes características:

- **Incertidumbre:** El desarrollo de los proyectos informáticos no se basa en procesos claramente definidos y no se aplican técnicas de gestión de proyectos. Estos puntos son características de la clasificación CMMI nivel 1.
- **Equipo pequeño de desarrollo:** La Gestión Artes Gráficas dispone de un equipo de desarrollo informático conformado por 4 personas, quienes están a cargo del desarrollo de todos los proyectos técnicos informáticos de esta Gestión.
- **Desarrollo en equipos:** Al utilizar estos proyectos técnicos nuevas herramientas tecnológicas es imprescindible que se realice transferencia del conocimiento de las mismas a todos los miembros del equipo. Las nuevas tecnologías exigen un alto grado de investigación por parte del equipo de desarrollo. En los proyecto se debe potenciar el acceso libre a la información y documentación, y favorecer la investigación.

- **Adaptación al cambio:** Valorar la respuesta al cambio, por encima del seguimiento de un plan: Anticipación y adaptación ante la planificación y control.

La complejidad de este tipo de proyectos hace que sea difícil planificar con exactitud las fases de desarrollo, lo que hace que éstas no existan como tal sino que se hace necesario ir desarrollando actividades en función de las necesidades, puesto que estas necesidades se la mayoría de ocasiones se tornan cambiantes en todo el proyecto.

- **Valoración de funcionalidad del software sobre la documentación exhaustiva:** A pesar de la importancia de documentación debido a que permite la transferencia de conocimiento, debe limitarse a la información que aporte valor directo al producto final.
- **Plan de Entregas:** La principal necesidad es lograr la satisfacción al cliente mediante la entrega de resultados continuos, de esta forma se favorece la retroalimentación del producto final durante su desarrollo, involucrando al cliente en toda las fases del proyecto.

1.1.1. COMPARACIÓN DE METODOLOGÍAS DE DESARROLLO VERSUS CARACTERÍSTICAS DE LOS PROYECTOS DE DESARROLLO INFORMÁTICO DE LA GESTION ARTES GRÁFICAS DEL IGM.

Al analizar las metodologías de desarrollo consideradas como candidatas al momento de ser implantadas en una empresa, surgen automáticamente, dos tipificaciones de metodologías: las metodologías ágiles versus las metodologías tradicionales versus las propiedades propias del proyecto informático y del equipo de desarrollo que la empresa dispone.

La implementación de cualquier tipo de proyecto a través de una metodología es completamente necesaria puesto que ésta permite tener un modelo en el cual se puede enfocar todos los proyectos en forma similar, además una metodología permite repetir los éxitos obtenidos y la implementación de mejoras para cada nuevo proyecto afrontado.

El objetivo principal de una metodología de desarrollo es aumentar la calidad del software por medio del control en todas las fases del proceso de desarrollo. En cualquier tipo de proyecto es importante generar el producto esperado en el tiempo esperado y con el presupuesto esperado.

Las metodologías para la implementación de proyectos de desarrollo de software constituyen un marco de trabajo mediante el cual un proyecto es planificado, estructurado y controlado en cada uno de sus procesos de desarrollo.

En el último tiempo el número de metodologías de desarrollo han aumentado considerablemente, siendo visible dos corrientes los métodos pesados y métodos ligeros, la principal diferencia entre estos es que los denominados métodos pesados se basan en la consecución del objetivo por medio del orden y la documentación exhaustiva, mientras que los denominados métodos ligeros

o ágiles se enfocan en mejorar la calidad de software por medio de la comunicación directa entre todas las personas que intervienen en el proceso. A través del tiempo y conforme el avance de la tecnología, las metodologías han ido desarrollándose y diferenciándose por sus fortalezas y debilidades, así como también por la especialización en tipos de proyectos, es decir han ido apareciendo metodologías que ofrecen mejores resultados al aplicarlas en proyectos de desarrollo de software que cumplen con ciertas características, lo que hace que ciertas metodologías ofrezcan mejores resultados que otras.

Las metodologías de desarrollo de software han demostrado ser efectivas en un gran número de proyectos, pero sin embargo en muchos otros no han mostrado resultados muy favorables. Una forma de atacar las debilidades identificadas en estas metodologías tradicionales es enfocarse en los puntos deficientes y fortalecerlos, pero esto podría derivar en producir una metodología compleja que resulte muy difícil implementarla por un equipo de desarrollo en un proyecto real. Otra forma de afrontar las deficiencias de las metodologías tradicionales es centrarse y poner énfasis en dimensiones que se consideren más importantes para la consecución de productos de calidad, estas dimensiones pueden ser por ejemplo el factor humano, el producto de software final en sí, o en la inclusión y participación del usuario final o cliente durante el proceso de desarrollo, etc.

En el presente trabajo se realiza el análisis mediante cuadro comparativo entre las características generales de las metodologías tradicionales de desarrollo versus características de las metodologías ágiles y versus las características de los proyectos de desarrollo informático de la Gestión Artes Gráficas del IGM, con el objeto de seleccionar en forma adecuada el tipo de metodología que se acople de mejor manera al desarrollo de los proyectos mencionados con mejores resultados en su consecución.

En la tabla 1.1 se muestra las diferencias existentes entre las metodologías de desarrollo tradicionales y metodologías ágiles versus la caracterización de los proyectos informáticos de Artes Gráficas en el IGM.

Metodologías Tradicionales	Metodologías Ágiles	Proyectos y equipo de desarrollo en la G. Artes Gráficas IGM
Requerimientos perfectamente definidos y estables, con cierta resistencia a los cambios.	Preparadas para afrontar requerimientos cambiantes a lo largo del proyecto.	Requerimientos cambiantes a lo largo del proyecto.
El cliente interactúa esporádicamente con el equipo de desarrollo mediante reuniones.	El cliente debe formar parte del equipo de desarrollo.	El cliente es involucrado en toda la fase de desarrollo.
Equipos medianos a grandes de desarrollo experiencia y conocimiento.	Se forman células de desarrollo, es decir equipos pequeños.	Se cuenta con recursos limitados en cuanto a desarrolladores.
Focalización en documentación, planificación y definición de procesos.	Focalización en la funcionalidad del producto.	Se debe proporcionar el software con la funcionalidad correcta en el tiempo establecido.
Desarrollo individual con roles y responsabilidades estrictas.	Transferencia de conocimiento: Programación en pares y conocimiento colectivo.	Se cuenta únicamente con dos desarrolladores para el desarrollo de estos proyectos.
Énfasis en el uso exhaustivo de documentación durante todo el ciclo del proyecto.	Énfasis en la capacidad de respuesta frente a variación de requerimientos.	Propenso a variación de requerimientos durante el desarrollo.
Seguimiento exhaustivo de un plan de proyecto definido en la etapa inicial.	Flexible a cambios durante la fase de desarrollo	Propenso a variación de requerimientos durante el desarrollo.
Toma de decisiones por parte del cliente al inicio del proyecto	Incertidumbre, donde el entorno es volátil, cuando los requisitos no se conocen con exactitud.	Proyectos inciertos.
Resultados visibles en la fase de implementación	Resultados visibles periódicamente en base a un plan de entregas.	Plan de entregas.

Tabla 1.1 Comparación Met de Desarrollo vs Características Proy. IGM

Referencia: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

Modificado por: M. Játiva, J. Carrera

De acuerdo a las características mostradas en la tabla 1.1, se puede concluir que los proyectos Informáticos de la Gestión Artes Gráficas del IGM presentan características que son manejadas de mejor forma con metodologías de desarrollo ágil. Es necesario enfocarse en la mejora de la calidad en la funcionalidad del software con tiempos de desarrollo reducidos y equipos de desarrollo pequeños, este proceso necesariamente debe contar con el involucramiento directo de todas las personas que intervienen en todas las fases del desarrollo, incluido el cliente o usuario de la aplicación. Adicionalmente las metodologías ágiles se deben aplicar en proyectos donde exista incertidumbre, donde el entorno es volátil y cuando los requisitos no se conocen con exactitud mientras que las metodologías tradicionales obligan al cliente a tomar las decisiones al inicio del proyecto

1.1.2. METODOLOGÍAS DE DESARROLLO ÁGIL

Las metodologías de desarrollo ágil pretenden ofrecer una alternativa a los procesos de desarrollo de software tradicionales, los cuales se caracterizan por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades del desarrollo.

Una metodología de desarrollo ágil valora y hace mayor énfasis en los siguientes puntos:

- A las interacciones entre los individuos que forman parte del equipo de desarrollo sobre el proceso y las herramientas. Cada persona constituye un factor muy importante para el éxito de un proyecto software.
- Es más importante desarrollar software completamente funcional antes que enfocarse en la elaboración de la documentación. Se debe producir únicamente los documentos que sean necesarios, estos deben ser cortos precisos.
- Constante colaboración e involucramiento del cliente con el equipo de desarrollo. Esta interacción asegurará el éxito del proyecto.

- Capacidad de respuesta frente a cambios, la planificación debe ser flexible y abierta.

De los principios anteriores se derivan 12 puntos que constituyen la filosofía de las metodologías de desarrollo ágil:

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

1.1.2.1. METODOLOGÍA DE DESARROLLO DYNAMIC SYSTEMS DEVELOPMENT METHOD(DSDM)

Definición: Planteada por un consorcio en enero de 1994, con el objetivo de generar una metodología independiente de las herramientas para ser utilizada en proyectos tipo RAD(Rapid ApplicationDevelopment).

La metodología DSDM contempla el:

- Definir un proceso iterativo e incremental involucrando al usuario en el trabajo del equipo de desarrollo.
- Presentar cinco fases, estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción e implementación.
- Realimentarse durante el proceso de desarrollo, esto es necesario para converger a la correcta solución del negocio.
- El equipo de desarrollo DSDM debe tener suficiente poder para la toma de decisiones.
- Tener entrega frecuente de productos.
- Los cambios durante el desarrollo del proyecto son reversibles.
- Los requerimientos deben ser especificados a un alto nivel.
- Se debe integrar procedimientos de pruebas durante todo ciclo de vida.

En la metodología DSDM el estudio de la factibilidad consiste en determinar si la metodología será ajustable al proyecto. En la fase de estudio del negocio se involucra al cliente desde los inicios de tal forma que el entendimiento del negocio a automatizar sea entendido en forma clara. El estudio y entendimiento de los requerimientos constituyen la base para el inicio del desarrollo definiendo las propiedades de alto nivel que el software deberá contener. A continuación se inician las iteraciones, en estas se considera a detalle las propiedades que se han identificado en la fase anterior para posteriormente generar el diseño e iniciar la construcción de

los componentes de software para concluir con la implantación del sistema con la validación y aceptación del cliente o usuario final.

Requisitos: La determinación de la aplicabilidad de la metodología DSDM debe ser resuelta mediante el análisis de los siguientes puntos:

- Complejidad computacional de la aplicación.
- Visibilidad de la funcionalidad de la aplicación en la interfaz de usuario.
- Tamaño de la aplicación y posibilidad de ser patrocinada en componentes funcionales más pequeños.
- Acotación en el tiempo del proyecto.
- Definición clara de usuarios.

Los proyectos que califiquen para la aplicabilidad de la metodología deben poseer las siguientes características en función al análisis de los puntos mencionados anteriormente:

- Proyectos de baja o mediana complejidad.
- La interactividad y funcionalidad de la aplicación debe ser visible en la interfaz de usuario.
- Aplicación de no gran tamaño o de ser el caso posibilidad de ser patrocinada en componentes funcionales pequeños.
- Disponibilidad de un periodo determinado de tiempo para el desarrollo del proyecto.
- El proyecto puede definir claramente los usuarios finales y estos se encuentran directamente comprometidos en el desarrollo de este.

La metodología sugiere la generación de un conjunto de modelos que deben ser definidos previo el inicio del desarrollo y que son necesarios para la progresión de la entrega del software y facilidad en el mantenimiento, estos modelos deben ser revisados en cada iteración.

Cada iteración en DSDM consta de una fase de investigación, una fase de refinamiento, y una última fase de consolidación.

En la fase de investigación se fijan los objetivos y los entregables de la iteración.

La fase de refinamiento consiste en la construcción de los modelos planificados.

En la fase de consolidación se debe completar los entregables verificando la calidad de estos, en esta fase se debe generar un hito donde se demuestra que se satisficieron los requerimientos de calidad definidos en la fase de investigación.

Las características de la metodología DSDM son las siguientes:

- El involucramiento del usuario es imperativo.
- Los equipos de DSDM deben tener el poder de tomar decisiones.
- El foco está puesto en la entrega frecuente de productos.
- La conformidad con los propósitos del negocio es el criterio esencial para la aceptación de los entregables.
- El desarrollo iterativo e incremental es necesario para converger hacia una correcta solución del negocio.
- Todos los cambios durante el desarrollo son reversibles.
- Los requerimientos están especificados a un alto nivel.
- El testing es integrado a través del ciclo de vida.
- Un enfoque colaborativo y cooperativo entre todos los interesados es esencial.

1.1.2.2. METODOLOGÍA DE DESARROLLO ÁGIL FEATURE DRIVEN DEVELOPMENT FDD

Esta metodología se basa en la definición de funcionalidades que deben ser implementadas en el sistema. FDD está pensado para proyectos con duración relativamente cortos, menores a un año.

En esta metodología se presenta una jerarquía de funcionalidades, la primera en la jerarquía agrupa las propiedades relacionadas con aspectos en común del negocio.

Una de las principales ventajas de trabajar centrado en las funcionalidades del software es el de formar y fomentar un dialogo fluido entre los clientes y desarrolladores y que ambos posean un modelo común del negocio.

FDD constituye un proceso iterativo con iteraciones cortas (menores a dos semanas) obteniendo como resultado un software funcional que la dirección de la empresa cliente puede ver y monitorizar.

Las iteraciones son definidas en base a las funcionalidades, un proyecto abordado con FDD presenta cinco fases:

1. Desarrollo de un modelo general.
2. Construcción de la lista de funcionalidades.
3. Plan de entregables en base a las funcionalidades a implementar.
4. Diseñar en base a las funcionalidades.
5. Implementar en base a las funcionalidades.

La figura 1.1 muestra las fases de la Metodología de desarrollo FDD.

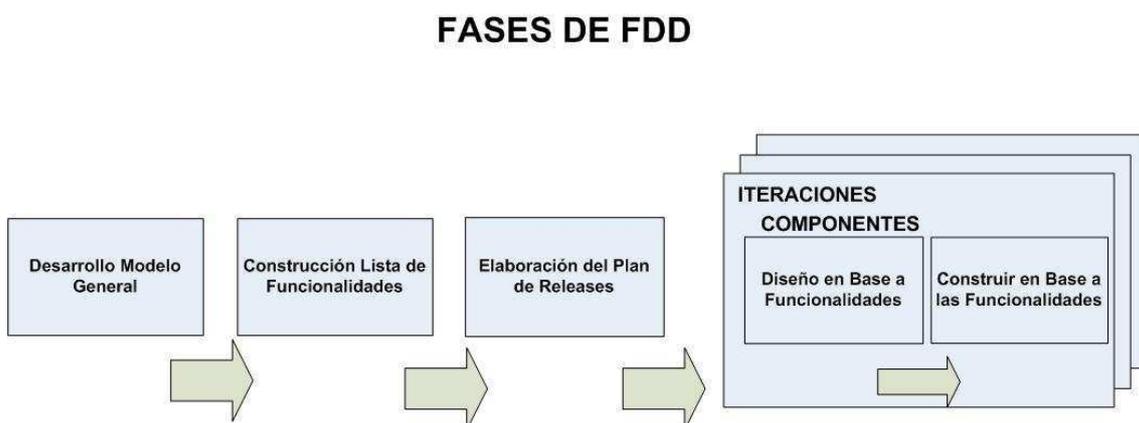


Figura 1.1: Fases Metodología de desarrollo FDD

Referencia: *Métodos de Desarrollo ágil de Software Online*

<https://sites.google.com/site/ingsoportelologico/home/feature-driven-development-fdd>

El trabajo de modelado como el de desarrollo es realizado en grupo, pero siempre se debe contar con la presencia de un jefe o arquitecto de desarrolladores, en función a la fase en la que el proyecto se encuentre.

Las funcionalidades que se deben implementar son divididas entre los subgrupos del equipo, cada clase escrita tiene un propietario, y se restringe a que solo el propietario de la clase pueda cambiar el código de esta, esto da como resultado que un programador puede integrar varios subgrupos y estará implementando diferentes funcionalidades en el proyecto.

En la primera fase del FDD deben participar tanto expertos en el dominio del negocio como los desarrolladores, mediante la colaboración de los dos grupos se genera un conocimiento global de la aplicación a construir y los primeros bosquejos de las funcionalidades del software.

La segunda fase de FDD es construir la lista de funcionalidades del software a desarrollar, para esta fase se toma el bosquejo elaborado en la fase anterior refinando las funcionalidades incluidas para ubicarlas en una estructura organizada en forma jerárquica. El trabajo de desarrollo es organizado en función de priorización de las funcionalidades definidas, en caso de obtener actividades cuyo desarrollo implique más de dos semanas, estas deben ser particionadas para ser ubicadas en iteraciones.

La tercera fase de FDD establece tiempos de desarrollo a partir de la lista priorizada de funcionalidades que se obtuvo en la fase anterior. Los líderes de proyecto deben delinear los hitos de finalización de cada iteración estableciendo responsabilidades.

La parte productiva en cuanto a desarrollo está contenida en las dos últimas fases de FDD, en estas fases se construye en forma incremental la aplicación. Mediante la utilización de lenguaje UML, se diseñan las clases atributos y métodos requeridos para la implementación de la funcionalidad de la aplicación.

La metodología FDD está orientada completamente a la funcionalidad del software sobre las tareas. La metodología recomienda organizar bloques de funcionalidades para ser construidos en forma incremental mediante iteraciones de dos semanas de duración, cada subgrupo de programación debe ser dirigido por un programador jefe.

Las características de la metodología FDD son las siguientes:

- No cubre todo el ciclo de vida del proyecto sino el diseño y construcción.
- Se considera apto para proyectos de misión crítica.
- Metodología de desarrollo de centrado en ciclos cortos.
- El modelo del dominio consta de diagramas de clases, relaciones métodos y atributos.
- Los métodos reflejan rasgos funcionales del software más no conveniencias de programación.

1.1.2.3. METODOLOGÍA DE DESARROLLO ÁGIL SCRUM

El término Scrum proviene de un “paper” titulado The New Product Development game de HirotakaTakeuchiy IkujiroNonaka. En rugby, un Scrum es una forma de recomenzar el juego, luego de una infracción accidental o después que la pelota salió de juego. La práctica de Scrum en el mundo del software incluye reuniones regulares diarias y cortas donde los miembros del equipo se juntan para comunicar su progreso. Debido a la similitud entre la pausa del juego (trabajo), y habiendo los jugadores (miembros del equipo) armado la reunión, ésta se conoce comúnmente como el DailyScrum.

Scrum es un framework de desarrollo ágil de software. Es un proceso de desarrollo de software iterativo y creciente que utiliza comúnmente entornos basados en el desarrollo ágil de software tales como trabajo en equipo, colaboración, delegación, creatividad, mejora continua y facilidad para realizar cambios.

El trabajo es estructurado en ciclos de trabajo llamados sprints, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente priorizados, llamados historias de usuarios, para que las características que sean desarrolladas primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto potencialmente funcional, distribuible y comerciable.

Dentro de las prácticas de Scrum se manejan además los conceptos de Sprint backlog que es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. El Burndown que es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, se puede ver el progreso del proyecto. Lo normal es que esta línea sea descendente.

1.1.2.3.1. CARACTERÍSTICAS SCRUM

1.1.2.3.1.1. EMPIRISMO

Se refiere al proceso continuo de inspeccionar y adaptar los avances realizados. Al final de cada iteración se demuestra al cliente el resultado real obtenido, lo que permite a los usuarios finales tomar decisiones en función de lo que observa y del contexto del proyecto en ese momento. Por otro lado, el equipo se sincroniza diariamente y realiza las adaptaciones necesarias. La inspección en tiempo real basados en datos actuales permiten responder rápidamente a condiciones cambiantes que se pueden presentar.

1.1.2.3.1.2. PRIORIZACIÓN

Consiste en determinar cuáles son las cosas más importantes a hacer primero, y entonces promover a hacerlas. Se propone priorizar los requisitos por valor para el cliente y coste de desarrollo en cada iteración. Al inicio de cada

iteración el cliente prioriza la lista de requisitos del producto o proyecto en función del valor que le aportan, su coste de desarrollo y los riesgos del proyecto, cambiando los requisitos previstos para reaccionar a cambios de contexto en el proyecto.

1.1.2.3.1.3. TIMEBOXING

La técnica del timebox consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar ciertas tareas, y hacer lo mejor que se pueda en ese intervalo. Timeboxing es un mecanismo simple para manejar la complejidad. No se puede imaginar la funcionalidad total de un sistema completo, pero se puede tomar un pequeño problema y en un corto espacio de tiempo, (una semana o un mes), trabajar en solucionar ese problema. Los resultados de esa acción guiarán entonces a una solución para el próximo problema, más grande, y nos dará más conocimiento sobre las necesidades del sistema en conjunto.

1.1.2.3.1.4. POTENCIACIÓN DE TRABAJO EN EQUIPO

Consiste en dar poder a pequeños equipos multidisciplinares para que puedan tomar decisiones importantes, necesarias para crear un producto de alta calidad, y manejar su propio proceso.

Se promueve que los equipos trabajen de una manera altamente interactiva y generativa, donde el producto emerge del diálogo continuo, de la exploración e iteración. La auto-organización funciona cuando hay objetivos y límites claros.

En el inicio de cada iteración el equipo selecciona los requisitos que se compromete a completar y entregar al final de la iteración (responsabilidad). El propio equipo se organiza (autoridad) identificando las tareas necesarias, su esfuerzo y auto asignándose cada miembro las tareas que se compromete a realizar.

1.1.2.3.1.5. COLABORACIÓN Y COMUNICACIÓN ENTRE EL EQUIPO Y CLIENTE

Los miembros del equipo deben sincronizar su trabajo diariamente y se debe cooperar en resolver problemas que pueden impedir conseguir el objetivo de la iteración. La comunicación y la adaptación a las diferentes necesidades entre los miembros del equipo son máximas (se van ajustando iteración a iteración), de manera que no se realizan tareas innecesarias y se evitan ineficiencias.

1.1.2.3.1.6. FLEXIBILIDAD

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea, los desafíos impredecibles no siempre pueden ser fácilmente enfrentados de una forma predictiva y planificada por tanto por cada iteración se permite la flexibilidad a cambios, de esta forma el equipo se evita caminar mucho tiempo por un camino equivocado que le obligue a realizar un gran esfuerzo para llegar al objetivo esperado.

1.1.2.3.1.7. REPLANIFICACIÓN AL INICIO DE CADA ITERACIÓN

Se asume que los cambios son parte natural del proyecto. Toda iteración comienza con una replanificación del proyecto. Esta replanificación no es traumática puesto que Scrum minimiza el número de objetivos/requisitos en que el equipo trabaja (WIP, Work In Progress) a los que caben en una iteración.

1.1.2.3.1.8. ORGANIZACIÓN DE EQUIPOS Y ROLES

Define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

1.1.2.3.1.9. DUEÑO DEL PRODUCTO

El dueño del producto representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El ProductOwner escribe historias de usuario, las prioriza, y las coloca en el ProductBacklog.

1.1.2.3.1.10. FACILITADOR

El Scrum es realizado por un facilitador, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El Facilitador no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El facilitador se asegura de que el proceso Scrum se utiliza como es debido. El facilitador es el que hace que las reglas se cumplan.

1.1.2.3.1.11. EQUIPO

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

A estos roles se les conoce como equipos comprometidos sin embargo un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados.

Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

1.1.2.3.2. VENTAJAS SCRUM

1.1.2.3.2.1. CAMBIO ORGANIZACIONAL

Con Scrum, las jerarquías de gerencia de las organizaciones tienden a ser niveladas y los equipos de desarrollo tienen más contacto directo e inmediato con los clientes. El ambiente de trabajo se vuelve menos “comandar-y-controlar” hacia un estilo más colaborativo. Se promueve el diálogo regular y abierto sobre la documentación extensiva, y el acuerdo negociado es preferido a los contratos de trabajo formales e impersonales.

Las decisiones se toman por consenso, más que por imposición de alguien de más arriba, y todo el conocimiento es compartido, de una manera transparente y sin recelos.

1.1.2.3.2.2. ENTREGA RÁPIDA DE RESULTADOS

Se realiza la entrega de los requisitos más prioritarios completados lo cual permite:

Gestionar y regular de las expectativas del cliente basada en resultados tangibles.

Resultados anticipados (time to market).

Flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, etc.

Gestión sistemática del Retorno de Inversión (ROI).

Mitigación sistemática de los riesgos del proyecto.

1.1.2.3.2.3. PRODUCTIVIDAD Y CALIDAD

De manera regular el equipo va mejorando y simplificando su forma de trabajar de esta manera se asegura la calidad del producto de manera sistemática y objetiva, a nivel de satisfacción del cliente, requisitos listos para ser utilizados y calidad interna del producto.

1.1.2.3.2.4. MEJORA CONTINUA

Cada iteración el equipo realiza una retrospectiva para analizar su manera de trabajar e identificar los obstáculos que le impiden avanzar al mejor ritmo posible.

1.1.2.3.2.5. COMUNICACIÓN DIARIA DEL EQUIPO Y COMPROMETIMIENTO DEL MISMO

Todo miembro del equipo debe conocer cómo el trabajo de los otros miembros impacta en el suyo y cuáles son las necesidades de los otros.

Las personas trabajan más enfocadas y de manera más eficiente cuando hay una fecha límite a corto plazo para entregar un resultado al que se han comprometido. La consciencia de esta limitación temporal favorece la priorización de las tareas y fuerza la toma de decisiones.

1.1.2.3.3. DESVENTAJAS SCRUM

Scrum está aún en la etapa de los que se acogen tempranamente a las nuevas ideas. Tomará muchos años para que la mayoría de las compañías reconozcan los beneficios de crear más lugares de trabajo, que ofrezcan mejores formas de trabajar. Sin un cambio de paradigma será muy complicado que haya éxito en la aplicación de Scrum, muchas compañías de software se mantendrán bajo el peso de sus procesos pesados, y fuerzas de trabajo sobrecargadas.

1.1.2.4. METODOLOGÍA DE DESARROLLO ÁGIL EXTREME PROGRAMING XP

No es más valioso el software complejo y difícil de mantener que aquel que es diseñado para ser elegante y simple. XP como toda metodología ágil de desarrollo intenta reducir la complejidad del software mediante la orientación del trabajo directamente al objetivo basándose en las relaciones interpersonales y en la velocidad de reacción.

1.1.2.4.1. OBJETIVOS DE LA METODOLOGÍA XP

El objetivo principal de esta metodología es el reducir el riesgo de fallo en el proceso por medio de la presencia de un representante disponible y competente del cliente en el equipo. Es decir que el principal objetivo que busca XP es conseguir la satisfacción del cliente.

El segundo objetivo de la metodología es el de potenciar al máximo el trabajo en equipo, los jefes de proyecto, desarrolladores y clientes son responsables de la implementación del software de calidad a través de la comunicación.

1.1.2.4.2. FASES DE LA METODOLOGÍA XP

Existen 4 fases en la metodología XP:

- Fase de Planificación.
- Fase de Diseño.
- Fase de Desarrollo.
- Fase de Pruebas.

1.1.2.4.2.1. FASE DE PLANIFICACIÓN

La fase de planificación en la metodología XP se plantea como un proceso de diálogo permanente entre la parte empresarial y la parte técnica del proyecto.

La parte empresarial define el alcance del proyecto, prioridades de desarrollo, funcionalidad de versiones y las fechas de entrega de cada una. La parte técnica tiene la responsabilidad de definir los tiempos requeridos para la implementación de las funcionalidades especificadas por el cliente, organizar la cultura de trabajo y de realizar la planificación detallada para cada versión.

La fase de Planificación en XP se compone de 5 procesos:

- Redacción de Historias de Usuarios.
- Elaboración del Plan de Entregas

- Control de la velocidad del proyecto.
- División del proyecto en iteraciones.
- Definición del Plan de Iteración.
- Rotación de personal.

1.1.2.4.2.1.1. REDACCIÓN DE HISTORIAS DE USUARIO

El propósito de las Historias de Usuario es similar al de los diagramas de casos de uso. Las Historias de Usuario son escritas por el propio cliente en función a su propia visión de las necesidades que el sistema deberá cubrir. Por tanto la historia de usuario son documentos con descripciones cortas en los que se detalla en lenguaje natural y sin terminología técnica alguna.

Las Historias de Usuario o User Stories constituyen la base del software a desarrollar. Las historias de usuario contienen la descripción de escenarios sobre el funcionamiento del software. A partir de éstas y de la definición de la arquitectura del software, se crea un plan de versiones entre el equipo de desarrollo y el cliente final.

Una Historia de Usuario no es igual al documento de especificación de requerimientos, su principal diferencia radica en el nivel de detalle que la historia de usuario tiene, en ésta se debe especificar el nivel de riesgo y cuánto tiempo tomará la implementación de dicha historia de usuario. Cuando el proyecto se encuentre en la fase de implementación, el equipo de desarrollo tendrá la posibilidad de acudir al cliente para ampliar detalles.

Por cada Historia de Usuario el equipo de desarrollo debe realizar estimaciones de tiempo de implementación en condiciones ideales, es decir en condiciones en las que los desarrolladores no tengan distracciones y asumiendo que se tiene totalmente claro que es lo que se debe implementar. Si el período estimado sobrepasa las tres semanas, se debe dividir en dos partes, si el tiempo es menor a una semana, entonces se debe unir dos o mas Historias de Usuario.

1.1.2.4.2.1.2. ELABORACIÓN DE PLAN DE ENTREGAS

A partir de las Historias de Usuario se procede a realizar un Plan de Entregas estimado.

El plan de entregas debe ser creado mediante una reunión en la que deben estar presentes la parte técnica y comercial y los usuarios. El cliente agrupará las Historias de Usuario previamente analizadas en orden de importancia. El Plan de Entregas es elaborado considerando el tiempo de desarrollo ideal (tiempo sin interrupciones) y el grado de importancia para el cliente.

1.1.2.4.2.1.3. CONTROL DE LA VELOCIDAD DEL PROYECTO

Este proceso es usado para determinar la medida de cuán rápido se está desarrollando. Esta medida puede ser establecida en base a tiempo y alcance.

Para el establecimiento de la velocidad del proyecto en base al tiempo se debe determinar cuántas Historias de Usuario pueden ser implementadas antes de una fecha dada. La velocidad del proyecto en base al alcance se obtiene de establecer cuanto tiempo será necesario para implementar un conjunto de historias determinado.

1.1.2.4.2.1.4. DIVISIÓN DEL PROYECTO EN ITERACIONES

La iteración comprende un intervalo de tiempo que va desde una a tres semanas de desarrollo del proyecto.

Es recomendable convocar a una reunión al principio de cada iteración para trazar un plan de iteración. No es posible implementar funcionalidades que no consten en la iteración, esto está totalmente prohibido.

En este proceso se usa la velocidad del proyecto para determinar si una iteración se encuentra sobrecargada, en tal caso el cliente deberá decidir según sus prioridades que historias de usuario son desplazadas a la siguiente iteración.

1.1.2.4.2.1.5. DEFINICIÓN DEL PLAN DE ITERACIÓN

En el Plan de Iteración se selecciona las Historias de Usuario que según el Plan de Entregas corresponderían a esta iteración. En este proceso se elige también las pruebas de aceptación fallidas que se corregirán.

En la definición del Plan de Iteración se transforma cada historia de usuario en tareas de desarrollo que deben ser cubiertas en el tiempo ideal de 3 días.

La figura 1.2 muestra el proceso de elaboración del Plan de Iteración en la metodología XP.

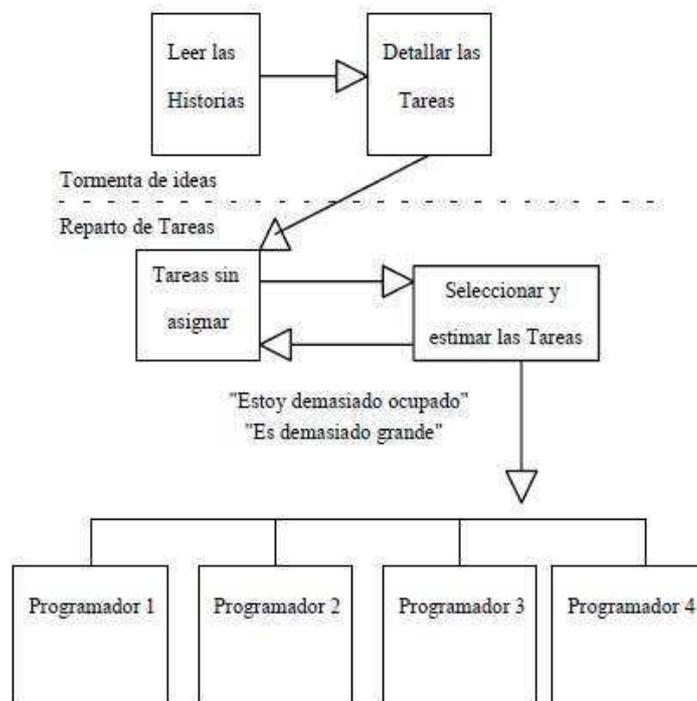


Figura 1.2:Elaboración del Plan de Iteración

Referencia: Programación Extrema Online

<http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQWdpbGVzL1Byb2dyYW1hY2lvbi9FeHRyZW1hLVhQLnBkZg%3D%3D&cidReset=true&cidReq=II0162004>

1.1.2.4.2.1.6. ROTACIÓN DE PERSONAL

Esta actividad evita que el riesgo de retraso para el proyecto sea grande en caso de dejar de contar con una persona determinada. Las rotaciones permiten que todos los miembros del equipo tenga un conocimiento de la funcionalidad del sistema, adicionalmente el hecho de que la asignación sea por parejas se cuenta con una ventaja en el entrenamiento de un nuevo miembro.

1.1.2.4.2.2. FASE DE DISEÑO

Se debe procurar la mayor simpleza en el diseño, es decir este debe ser lo más sencillo posible, si alguna parte de la implementación se torna dificultosa, se debe contemplar la posibilidad de replantear el diseño. En ocasiones obtener un diseño sencillo se puede tornar especialmente dificultoso.

Se debe evitar añadir funcionalidades en las primeras etapas, aun cuando se tenga la certeza absoluta de conocer su implementación. Es necesario siempre centrarse en los objetivos que se han trazado para ese momento. Solo se debe desarrollar código de lo que se ha fijado.

1.1.2.4.2.3. FASE DE DESARROLLO

En esta etapa se debe asegurar que se cumpla con las siguientes características.

- Disponibilidad del cliente
- Escritura del código según los estándares.
- Implementar unidad de pruebas previo el desarrollo.
- Programar todo el código por parejas.
- Una sola pareja es encargada de la integración del código.
- Integración frecuente con las versiones recientes.

- No trabajar más de 40 horas semanales.

1.1.2.4.2.4. FASE DE PRUEBAS

Todo el código desarrollado debe ir acompañado de su unidad de pruebas, los test de pruebas constituyen un pilar básico de la metodología XP.

El descubrimiento de los errores lleva tiempo y es más peligroso si se deja esta actividad para el final del proyecto. Después de cada modificación se empleará un test para verificar que ese cambio no alteró la funcionalidad.

Las unidades de prueba deben ser creadas previo el desarrollo de código, esto garantiza su independencia y objetividad, puesto que si fueran creadas después, se utilizaría el código como un generador y los errores del código serían pasados a las unidades de prueba.

Las unidades de prueba constituyen una herramienta de desarrollo y no pueden ser pasadas por alto por ningún motivo.

En esta fase se deben ejecutar pruebas de aceptación, y publicar sus resultados con regularidad. Las pruebas de aceptación son definidas en las historias de usuario, el cliente o usuario especificará los componentes a probar.

Es responsabilidad del cliente el verificar la corrección de las pruebas de aceptación, una historia de usuario no es completada hasta que no supere las pruebas de aceptación.

La publicación de los resultados de las pruebas deben ser realizadas lo más rápido posible, de tal forma que los desarrolladores puedan efectuar los cambios que sean necesarios.

Las pruebas de aceptación o pruebas de funcionalidad, constituyen una garantía de que los requerimientos fijados por los usuarios han sido reflejados en el sistema.

1.2. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO PARA PROYECTOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM.

Puesto que todas las metodologías de desarrollo persiguen el objetivo de mejorar la calidad en el software producido, se podría pensar que no es posible comparar las diferentes metodologías de desarrollo que existen, y que se puede dejar a criterio personal la selección de una u otra, pero se debe tomar en cuenta varios aspectos y consideraciones externas a la metodología que influyen directa o indirectamente en el proceso de desarrollo. En la vida real de una empresa en cuya misión no se considera la producción de software, estos aspectos son fundamentales al momento de decidirse por una u otra metodología.

Para la selección de la metodología ágil a utilizar en la elaboración de este trabajo, se ha considerado el estudio realizado por el Instituto de Ingeniería de Software de la Universidad “The Carnegie Mellon University”, en el cual de acuerdo a sus experiencias en desarrollo se asigna valores numéricos a diferentes características de las metodologías, para de esta forma determinar la más “ágil”.

La tabla 1.2, muestra los resultados obtenidos en el análisis elaborado por “The Carnegie Mellon University”.

Características de la Metodología	DSDM	FDD	SCRUM	XP
Resultados	4	4	5	5
Simplicidad	3	5	5	5
Adaptabilidad	3	3	4	3
Excelencia Técnica	4	4	3	4
Prácticas de colaboración	4	3	4	5
Media Total	3.6	3.8	4.2	4.4

Tabla 1.2 Comparación de metodologías ágiles
Referencia: Carnegie Mellon University

Adicionalmente en la selección de la metodología se debe considerar aspectos propios de la realidad de la Gestión Artes Gráficas del Instituto Geográfico Militar. El equipo de desarrollo con el que se cuenta comprende de cuatro desarrolladores los que se encargan de afrontar todo tipo de proyecto productivo que implique desarrollo de software, puesto que el IGM es una empresa del Estado no es posible la asignación de más recursos humanos para el efecto. Todo el proceso de desarrollo es afrontado por los cuatro recursos asignados.

Por las características y ventajas mostradas en el estudio de “The Carnegie Mellon University” sobre las metodologías de desarrollo XP y SCRUM y por las condiciones antes mencionadas en cuanto a recursos humanos para formar equipos de desarrollo, se ve conveniente el tomar procedimientos de las metodología SCRUM Y XP para el desarrollo de los proyectos técnicos de la Gestión Artes Gráficas, haciendo énfasis en una de las principales características del XP como es el desarrollo en pares y en las iteraciones de SCRUM.

1.3. MODELOS DE CALIDAD PARA PROYECTOS INFORMÁTICOS DE LA GESTIÓN ARTES GRÁFICAS DEL IGM.

Según R. Presuman 1998, calidad es “Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario”

La implementación o adopción de un sistema para la gestión de la calidad debe ser una decisión estratégica para una organización. El diseño e implementación de un sistema para la gestión de la calidad debe considerar aspectos propios de la organización como el entorno en el que se desenvuelve, sus objetivos, los productos que elaboran, procesos o el tamaño y estructura de la organización.

La evaluación de la calidad de un producto de software es diferente en relación a evaluar la calidad en otros productos de fabricación industrial. La dificultad presente en el software se debe a las características abstractas que este posee, por tal razón en el ámbito informático no basta con considerar la calidad de un producto una vez que este haya finalizado, puesto que la solución a posibles problemas que se puedan presentar puede ser costosa y en ocasiones inalcanzable. Con la implementación de un proceso de mejora continua en los procesos de desarrollo se puede asegurar la calidad del producto durante y después de la ejecución del proyecto.

Los estándares o metodologías de calidad definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad por este motivo todas las metodologías y herramientas tienen un único fin producir software de calidad.

El presente trabajo se orientará a incorporar estándares de calidad apropiados a todas las fases de desarrollo de software y de ser el caso modificar o incorporar algunas prácticas con la finalidad de lograr cumplir con las cláusulas que la metodología de calidad seleccionada proponga.

En el desarrollo de software existen numerosas propuestas metodológicas que inciden en las fases del proceso de desarrollo. La filosofía de algunas metodologías es dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque ha dado resultados favorables en proyectos cuyos requisitos son cambiantes y en aquellos en los que se necesita que los tiempos de desarrollo sean cortos pero sin perder la calidad del producto de software final.

Los factores que determinan la calidad del software según McCall son:

- Características operativas
- Capacidad de soportar los cambios
- Adaptabilidad a nuevos entornos

Características operativas

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

Capacidad de soportar los cambios

- Facilidad de mantenimiento. ¿Puedo corregirlo?
- Flexibilidad. ¿Puedo cambiarlo?
- Facilidad de prueba. ¿Puedo probarlo?

Adaptabilidad a nuevos entornos

- Portabilidad. ¿Podré usarlo en otra máquina?

- Reusabilidad. ¿Podré reutilizar alguna parte del software?
- Interoperabilidad. ¿Podré hacerlo interactuar con otro sistema?

Los estándares y modelos de evaluación y mejora de los procesos software han surgido como una guía para la gestión de calidad de los productos de software, los modelos entre los que se decidirá la adopción de uno se tiene:

- ISO 9001:2008
- CMMI

1.3.1.ISO 9001:2008

Los requisitos de esta norma son genéricos y lo que se busca es que sean aplicables a cualquier organización independientemente de su tamaño, tipo, producto o servicio suministrado. Para la aplicación de la norma, es necesario que la organización defina un enfoque basado en procesos de todas sus actividades.

La gestión de las actividades interrelacionadas entre si hace que el funcionamiento de una empresa sea eficaz. La actividad o actividades que transforman los recursos utilizados como entradas en resultados constituyen los procesos.

El enfoque basado en procesos en una organización es la adopción de un sistema que gestione cada uno de estos para el aseguramiento de que cada proceso produzca el resultado deseado y esperado.

La norma ISO 9001 determina los requisitos necesarios para un sistema de gestión de la calidad a utilizar internamente en una organización. Esta norma se centra en el nivel de eficacia del sistema de gestión de calidad, esto conlleva a la satisfacción de los requisitos de los clientes.

Los clientes juegan un papel importante al definir los requisitos como elementos de entrada.

Esta norma contempla el aseguramiento de la calidad incluyendo la necesidad de que las empresas demuestren su capacidad para satisfacer al cliente y mejorar sus procesos continuamente. La norma actúa sobre el modelo de procesos que identifica la empresa como un conjunto de procesos que se interrelacionan entre si. En el se propone la identificación sistemática y la gestión de los procesos como el elemento clave en la gestión de la calidad de cualquier empresa.

Un enfoque basado en procesos enfatiza la importancia de:

- La comprensión y cumplimiento de los requisitos definidos.
- Definición de los procesos como generadores de valor.
- Aseguramiento de la obtención de los resultados esperados con procesos eficaces.
- Medición y mejora continua de cada proceso.

La figura 1.3, muestra el modelo de un sistema de gestión de calidad basado en procesos, este modelo cubre los requisitos de esta norma, sin mostrar los procesos en una forma detallada.

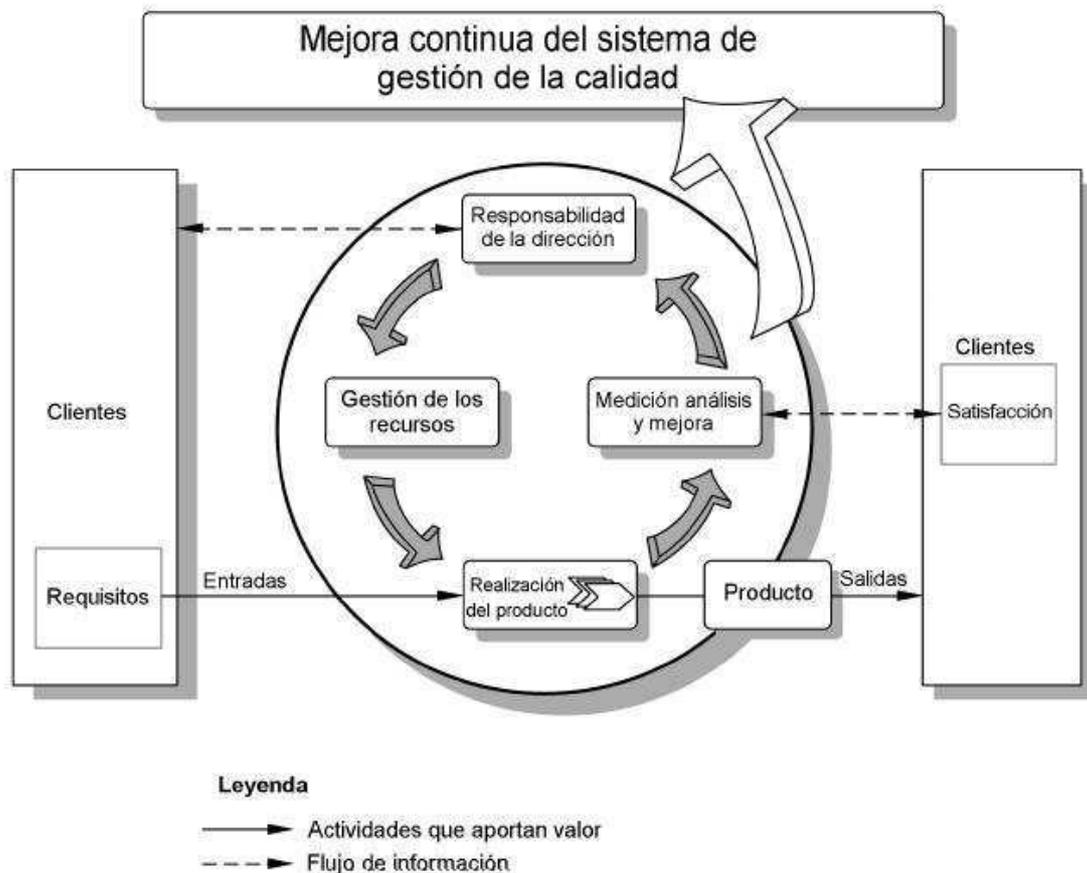


Figura 1.3: Mejora Continua del Sistema de Gestión de Calidad

Referencia: Evolución de los Sistemas de Calidad Online

http://www.vet-uy.com/articulos/artic_prof/017/017prof.htm

El control continuo sobre los vínculos existentes entre procesos individuales dentro del sistema de procesos, es una ventaja de manejar un enfoque basado en procesos dentro de la organización.

1.3.2. CMMI

Según Mark C. Paulk "CMMI es una aplicación de sentido común de los conceptos de gestión de procesos y mejora de la calidad al desarrollo y mantenimiento del software".

La creciente necesidad, sumada a décadas de promesas incumplidas en cuanto a calidad, costos y cumplimiento en el desarrollo de software, condujo al Instituto de Ingeniería de Software de los Estados Unidos a desarrollar el

modelo CMMI (CapabilityMaturityModelIntegration).

En principio creado para evaluar y mejorar la capacidad de los contratistas de software del Departamento de Defensa de los Estados Unidos, el modelo CMMI se convirtió a través de los años en el más alto estándar de ingeniería en el mundo para todo tipo de compañías. Está fundamentado en prácticas reales de las compañías más avanzadas del planeta, y refleja el estado del arte en procesos de desarrollo de software.

El CMMI está se compone de varias prácticas claves agrupadas en áreas y distribuidas en una jerarquía de cinco niveles, a través de los cuales una organización progresivamente alcanza mayor calidad, productividad y menores costos en el desarrollo de software. Los niveles progresan desde el 1, que representa el estado caótico, hasta el nivel 5, que representa el estado de optimización continua. Una organización en nivel 1, en el cual se encuentran la mayoría de los grupos de desarrollo en el mundo, produce software utilizando una aproximación de tanteo y error. Una organización en nivel 5 utiliza las mejores prácticas de ingeniería disponibles en el planeta, hace uso de procesos controlados, medibles y en continuo mejoramiento. Es altamente madura y sistemáticamente está en capacidad de producir software de alta calidad.

CMMI es un esquema que representa un camino de mejoramientos, permite determinar la madurez y evaluar las capacidades de las organizaciones que desarrollan software, CMMI es recomendado para organizaciones que quieren incrementar la capacidad de su proceso de desarrollo por tanto CMMI estudia:

- Los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de cinco niveles.
- La madurez de un proceso es un indicador de la capacidad para construir un software de calidad.

- Presenta un modelo para la mejora de las organizaciones.
- Obliga a una revisión constante.

La figura 1.4 muestra la estructura que sigue CMMI en cada nivel de madurez.

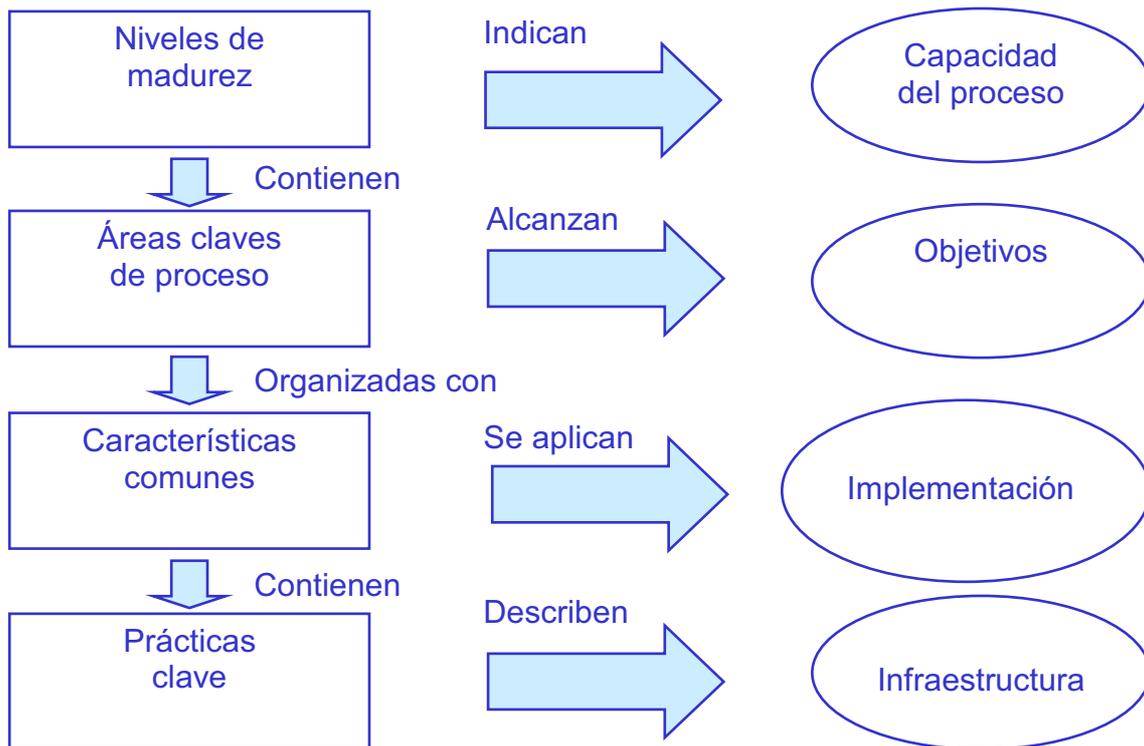


Figura 1.4: Estructura CMMI en un nivel de madurez
Referencia: Calidad en Ingeniería de Software Online

<http://dmi.uib.es/~bbuades/calidad/calidad.PPT>

1.3.2.1. NIVELES CMMI

1.3.2.1.1. Nivel 1. Inicial. En este nivel, los procesos y métodos de ingeniería no se encuentran definidos. Por esa razón, los proyectos son adelantados de manera incoherente, incontrolada y poco profesional. El éxito es eventual y depende del comportamiento heroico de algunos individuos, cuando estos poseen algún nivel de conocimiento. La mayoría de los grupos de desarrollo de software en el mundo operan a este nivel.

1.3.2.1.2. Nivel 2. Repetible. Se establecen algunos procesos y métodos de ingeniería a nivel de proyectos, aún incipientes.

1.3.2.1.3. Nivel 3. Definido. Los procesos, actividades y métodos relacionados con la ingeniería y administración de proyectos se encuentran documentados, estandarizados y construidos alrededor de un marco integrado para toda la compañía. Todos los integrantes de la organización los utilizan en su trabajo diario.

1.3.2.1.4. Nivel 4. Administrado. La compañía opera bajo Control Estadístico de Procesos, tanto en procesos como en productos. Los resultados de los procesos y la calidad de los productos son predecibles, y se controlan siguiendo las técnicas inicialmente publicadas por Deming, Crosby y Juran, técnicas que se han convertido en herramienta fundamental para las compañías de alta capacidad en el mundo.

1.3.2.1.5. Nivel 5. Optimización. En este nivel, las organizaciones se encuentran en un proceso de mejoramiento continuo. Todos los procesos y técnicas modernas están en pie, lo mismo que la administración cuantitativa. Las organizaciones se enfocan en el mejoramiento a través de técnicas y procesos de prevención de defectos, cambios en tecnología y cambios en procesos. Menos del 0.1% de las organizaciones en el mundo se encuentran en este nivel de madurez.

La figura 1.5 hace referencia a los niveles de madurez de CMMI.

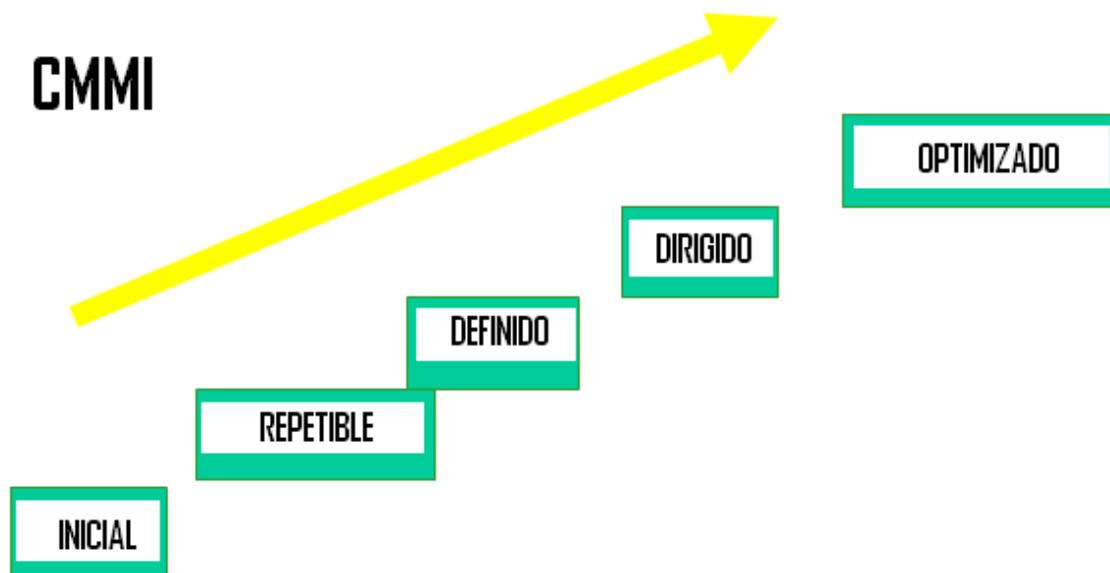


Figura 1.5: Niveles de Madurez CMMI

Traducido por: M. Játiva, J. Carrera

Referencia: Calidad en Ingeniería de Software Online

<http://dmi.uib.es/~bbuades/calidad/calidad.PPT>

Los principales beneficios que provee CMM son: mejorar la calidad de los productos, aumentar tiempo de respuesta al mercado e incrementara productividad de la organización.

1.4. SELECCIÓN DEL MODELO DE CALIDAD PARA PROYECTOS INFORMÁTICOS DE LA GESTION ARTES GRÁFICAS DEL IGM.

La norma ISO 90001 contempla requisitos genéricos que lo que buscan es ser aplicables a cualquier organización independientemente de su tamaño, tipo, producto o servicio suministrado.

CMMI califica la madurez de una empresa específicamente en sus procesos de desarrollo de software en una jerarquía de cinco niveles, a través de los cuales una organización progresivamente alcanza mayor calidad, productividad y menores costos en el desarrollo de software.

El modelo CMMI cuenta con aspectos estrechamente ligados a organizaciones complejas integradas por un gran número de personas, sin embargo uno de los objetivos de este trabajo es la incorporación de estándares de calidad a las fases de desarrollo de software en los proyectos informáticos técnicos de la Gestión Artes Gráficas del IGM considerando que el equipo de desarrollo y el tiempo de ejecución para estos proyectos son pequeños, siendo ésta la realidad del IGM y de gran número de organizaciones en el Ecuador. .

La adopción de CMMI en una empresa con las características del IGM presenta aspectos a resolver como:

- Documentación excesiva del proceso de desarrollo.
- Planificación, organización de KPAs orientadas a grandes organizaciones.
- Recursos limitados para el desarrollo de software.
- Costos altos en entrenamiento.

En equipos pequeños, la calidad, las habilidades y experiencia de estos juegan un papel fundamental en la obtención de productos de calidad. El grupo no puede dedicar mucho tiempo a trabajo administrativo, el tiempo debe ser enfocado en gran parte a trabajo de diseño, programación y pruebas de producto en construcción.

Existen algunas nuevas propuestas y modelos para ser aplicadas en pequeñas empresas. En México se plantea el modelo Moprosoft, este fue elaborado por la Asociación Mexicana para la Calidad de ingeniería de software a través de la Universidad Nacional Autónoma de México. Este modelo de proceso permite a las pequeñas y medianas empresas de desarrollo de software el acceso a las prácticas de Ingeniería de Software de “clase mundial” y se encuentra fundamentado en los niveles 2 y 3 de SW-CMM, ISO 9000 e ISO/IEC TR 15504.

MoProSoft está basado en procesos y considera los tres niveles básicos de la estructura de una organización que son:

- Alta Dirección.

- Gestión
- Operación.

La Figura 1.6 Muestra la arquitectura de los Procesos MoProSoft.

Arquitectura de Procesos de MoProSoft

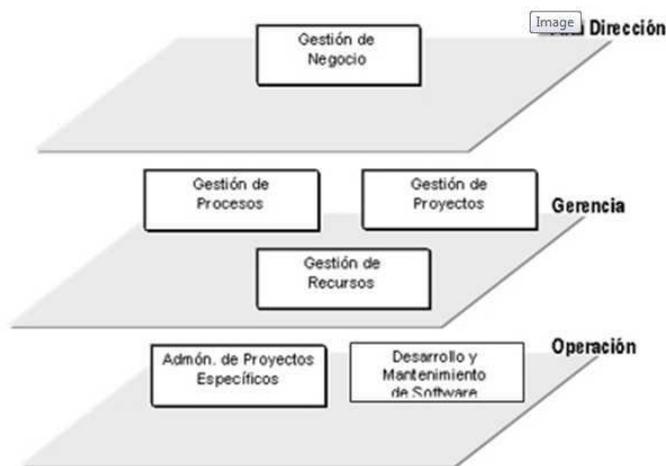


Figura 1.6: Arquitectura MoProSoft

Referencia: Modelo MoProsoft Online

<http://materias.utags.edu.mx/claroline/backends/download.php?url=L0FydGJldWxvcy9QMI8tX0FycXVpdG>

VjdHVyYV9Nb1Byb1NvZnZucGRm&cidReset=true&cidReq=CALS_IVET

El modelo tiene por objetivo apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua.

La categoría de Alta Dirección, está compuesta por el proceso de Gestión de Negocios, y se compone de la planificación estratégica, la preparación para la realización de la estrategia y la valoración y mejora continua de la organización.

En la categoría de Gestión, los procesos incluidos son los de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos (Humanos y Ambiente de trabajo, Bienes, Servicios en infraestructura y Conocimiento de la

Organización). Esta categoría de Gestión aborda las prácticas de gestión de procesos y recursos en forma coherente con los lineamientos de la Alta Dirección, proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados ala Categoría de Alta Dirección.

La última Categoría, la de Operación, aborda las prácticas de los proyectos de desarrollo y mantenimiento de software, y se compone de los procesos de Administración de Proyectos Específicos Desarrollo y Mantenimiento de Software.

En la figura 1.7 se puede observar los procesos planteados en el modelo MoproSoft.

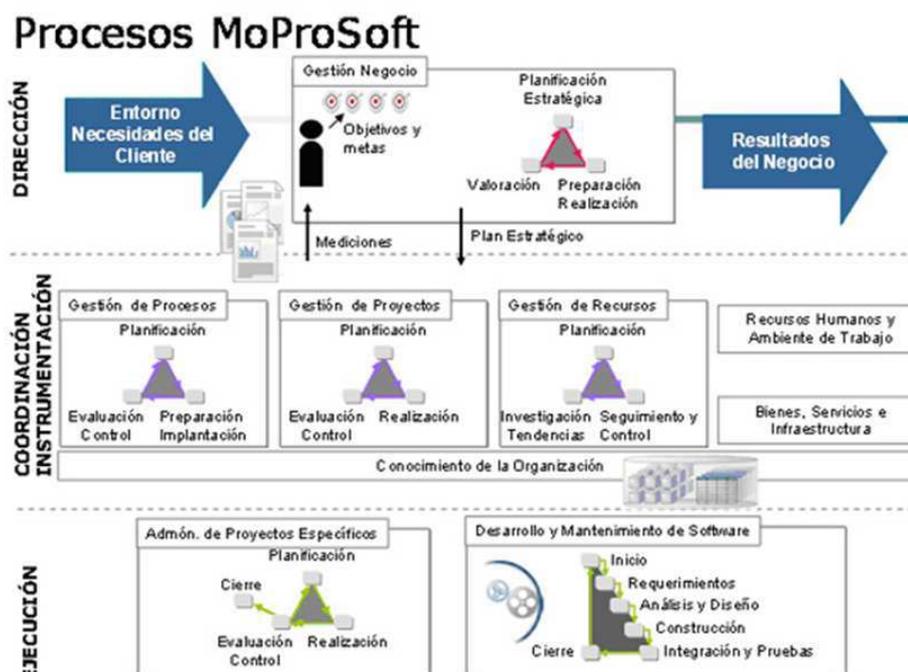


Figura 1.7: Procesos MoProSoft
Referencia: Modelo MoProsoft Online

http://materias.utags.edu.mx/claroline/backends/download.php?url=L0FydGJldWxvcy9QMl8tX0FycXVpdG VjdHVyYV9Nb1Byb1NvZnZnQucGRm&cidReset=true&cidReq=CALS_IVET

Por otro lado La Universidad Autónoma de Aguascalientes propone un nuevo modelo para lo cual realizó un estudio que muestra la solución al desarrollo de software en equipos pequeños, los cuáles pueden considerarse con un máximo de 20 personas.

El modelo propuesto involucra el desarrollo de 12 actividades, estas actividades están conformadas como una estrategia para garantizar la implementación de CMM en la organización, así como obtener una mejora en el proceso de desarrollo, las 12 actividades propuestas son:

- Compromiso de la dirección en la consecución del proyecto.
- Establecimiento del SPEG (Software Process Engineering Group).
- Definición de meta de proyectos y organización.
- Creación de una guía para el aseguramiento de las metas.
- Plan de información sobre CMM en la organización.
- Desarrollo de un proceso de software común.
- Proceso trazado por CMM
- Plan y operación de entrenamiento en la organización.
- Colección de los datos y mejora del proceso.
- Desarrollo del proceso pequeño (ajustado).
- Una valoración de SQA (Software Quality Assurance), es decir el establecimiento del programa de aseguramiento de la calidad de software.
- Mejora continua.

La creación del modelo debe indiscutiblemente realizarse en función de la reducción del ciclo de desarrollo, introduciendo la metodología ágil de desarrollo en un contexto CMMI. En este trabajo se ha tomado la decisión de establecer las actividades que dictan y SCRUM y XP adaptándolas a la realidad de la Gestión Artes Gráficas del Instituto Geográfico Militar para en cubrir con las KPAs definidas en CMMI en los niveles de madurez 2 y 3. Se debe tomar en cuenta que el IGM no es una empresa cuyo objeto de negocio es el desarrollo de software por lo tanto es claro que la estructura organizacional de la institución no está diseñada para tal efecto, por tanto en las KPAs que hacen referencia a la estructura organizacional se hará una adaptación con los recursos que la Gestión Tecnológica posee, puesto que

cualquier modificación en la estructura organizacional del Instituto se encuentra fuera del alcance de los desarrolladores de este trabajo.

En el siguiente capítulo se procederá a definir las KPAs del nivel 2 y 3 de CMM y las actividades de la metodología ágil seleccionada que cubren con las mencionadas KPAs.

CAPÍTULO 2 . CONSTRUCCIÓN DE UN MODELO DE GESTIÓN DE CALIDAD PARA DESARROLLO DE PROYECTOS INFORMÁTICOS DE GAG. IGM.

Es posible juzgar los procesos de desarrollo de software de una organización mediante la utilización del Modelo de Capacidad de Madurez (CMMI), sin embargo un problema que se presenta frecuentemente es que este modelo se lo aplica asumiendo que sólo es funcional para procesos rígidos definidos en el desarrollo de software, lo que conduce a que proyectos de contextos dinámicos en los que generalmente los requerimientos son cambiantes, culminen en fracaso.

En este trabajo se pretende mostrar cómo es posible combinar una metodología ágil para desarrollo de proyectos informáticos de contexto cambiantes con CMMI, determinando el objetivo de cada Área Clave de Proceso (KPA) cuando se utiliza una metodología ágil para desarrollo.

Este trabajo es direccionado para entidades en los que los proyectos de desarrollo informático técnico que se caractericen por el establecimiento de tiempos de desarrollo muy reducidos, sin perder un nivel alto de calidad y satisfacción del usuario final ante el producto de software generado.

Los factores considerados claves para conseguir la reducción del proceso de desarrollo de software sin afectar la calidad son:

- Determinación de prioridades en actividades de equipos pequeños, enfatizando en la construcción del software.
- Cumplir los objetivos únicamente de las KPAs que se relacionan con la definición de procesos de proyectos cortos, eliminando KPAs no necesarias en organizaciones pequeñas ni en proyectos cortos.

- Disminución de documentación y formatos de control, de tal forma que se pueda facilitar la administración del proyecto.

Las tablas 2.1 y 2.2 muestran las KPAs especificadas para alcanzar el nivel de madurez 2 y 3 de CMMI respectivamente.

KPA	NOMBRE
KPA1	RQM Gestión de Requerimientos (Requirements Management)
KPA2	PP Planificación de Proyectos de Software (Project Planning)
KPA3	PMC Seguimiento y Supervisión de Proyectos (Project Monitoring and Control)
KPA4	SAM Administración y Contratación Proveedores (Supplier Agreement Management)
KPA5	MA Métricas y Análisis (Measurement and Analysis)
KPA6	PPQA Aseguramiento de Calidad Proceso y Producto (Process and Product Quality Assurance)
KPA7	CM Gestión de Configuración (Configuration Management)

Tabla 2.1: KPAs Nivel 2 CMMI Repetible
Elaborado por: M. Játiva, J. Carrera

KPA	DESCRIPCIÓN
KPA8	RD Desarrollo de Requerimientos. (Requirements Development)
KPA9	TS Solución Técnica (Technical Solution)
KPA10	PI Integración del Producto (Product Integration)

KPA11	VER Verificación (Verification)
KPA12	VAL Validación (Validation)
KPA13	OPF Enfoque en el Proceso Organizacional (OrganizationalProcessFocus)
KPA14	OPD Definición del Proceso Organizacional (OrganizationalProcessDefinition)
KPA15	OT Capacitación Organizacional. (Organizational Training)
KPA16	RSKM Gestión de Riesgos 8 Risk Management).
KPA17	DAR Análisis de Decisiones y Resoluciones (DecisionAnalysis and Resolution)
KPA18	Ambiente de Desarrollo Integrado Producto y Proceso.(Integrated Product and Process Development (IPPD) nvironment
KPA19	IPM Gestión Integrada del Proyecto (Integrated Project Management)
KPA20	IT Integración de Equipo (IntegratedTeaming)
KPA21	ISM Gestión Integrada de Proveedores (Integrated Supplier Management)
KPA22	OEI Integración del Ambiente Organizacional (OrganizationalEnvironmentforIntegration)

Tabla 2.2: KPAs Nivel 3 CMMI Definido
Elaborado por: M. Játiva, J. Carrera

Para la definición del Modelo de Gestión que se propone, se realizará una clasificación de las KPAs dentro de dos ámbitos, de Producto de Software (PS) y Organizacional (O) y además se establecerá una selección de las KPAs que son aplicables según el tipo de proyectos que en este trabajo se aborda.

Las KPAs que afectan directamente al proceso de desarrollo de software dentro del Modelo de Gestión propuesto corresponden al Ámbito de Producto de Software (PS), mientras que las KPAs que se refieren a información general de la organización estarán incluidas dentro del Ámbito Organizacional (O).

DETERMINACIÓN Y CLASIFICACIÓN DE KPAs NIVEL 2 CMMI APLICABLES AL MODELO DE GESTIÓN PROPUESTO

En la tabla 2.3 se presenta la clasificación de las KPAs del nivel 2 CMMI que son aplicables al modelo propuesto.

KPA	NOMBRE	ÁMBITO	APLICA
KPA1	RQM Gestión de Requerimientos (Requirements Management)	PS	SI
KPA2	PP Planificación de Proyectos de Software (Project Planning)	PS	SI
KPA3	PMC Seguimiento y Supervisión de Proyectos (Project Monitoring and Control)	PS	SI
KPA4	SAM Administración y Contratación Proveedores (Supplier Agreement Management)		NO
KPA5	MA Métricas y Análisis (Measurement and Analysis)		NO
KPA6	PPQA Aseguramiento de Calidad Proceso y Producto (Process and Product Quality Assurance)	PS	SI
KPA7	CM Gestión de Configuración (Configuration Management)	PS	SI

**Tabla 2.3: Selección y clasificación KPAs aplicables NIVEL 2 CMMI
Elaborado por: M. Játiva, J.Carrera**

2.1.1. KPAs NO APLICABLES NIVEL 2 CMMI

2.1.1.1. KPA 4: ADMINISTRACIÓN Y COTRATACIÓN DE PROVEEDORES

En el Nivel 2 de madurez de CMMI no aplica la KPA 4 (SAM) referente a la Administración y Contratación de Proveedores, puesto que este modelo está dirigido a empresas o instituciones cuyo objeto de negocio no es el desarrollo de software y por tanto la administración y contratación de proveedores es realizada por una Unidad específica y está fuera de la competencia y el alcance de la Unidad de Gestión Tecnológica de la empresa.

La KPA4 SAM se cubre también la gestión del software de subcontratación y para el caso de los proyectos abordados en este trabajo, tampoco es aplicable, puesto que todo el desarrollo debe ser realizado por el equipo de trabajo interno, tomando en cuenta especificaciones de requerimientos internos específicamente para equipos de hardware que la empresa utilice en sus proyectos, estos dispositivos de hardware pueden ser impresoras, cámaras fotográficas, lectores de huellas dactilares, etc.

Los requerimientos en el desarrollo de software para este tipo de proyectos son puntuales para la utilización de equipos de hardware y específicos para manejo de datos, por esta razón no se requiere la subcontratación de software, cada desarrollo puede requerir de librerías o componentes especiales propias del equipo de hardware, estos paquetes son provistos por el fabricante y tanto el software como la asistencia técnica necesaria es contemplada dentro del contrato de adquisición de equipos y de mantenimiento.

2.1.1.2. KPA 5: METRICAS Y ANÁLISIS

En este nivel 2 de madurez CMM Definido, tampoco es aplicable la KPA 5 (MA), referente a Métricas y Análisis tomando en cuenta que una de las principales características de los proyectos abordados en el presente trabajo es que la definición de requerimientos es la incertidumbre en los requerimientos, es decir que estos pueden cambiar durante el proceso de desarrollo y además se cuenta con tiempos de desarrollo cortos, los parámetros para toma de decisiones durante el proceso de desarrollo deberán ser realizados en base a la definición de los elementos de la Pila del Producto y congelación Pila de

Sprint en la reunión de planificación del mismo. La forma en que se realizará estas acciones es especificada más adelante en el presente documento en el desarrollo del Modelo de Gestión propuesto.

2.2. DETERMINACIÓN Y CLASIFICACIÓN DE KPAs NIVEL 3 CMMI APLICABLES AL MODELO DE GESTIÓN PROPUESTO

En la tabla 2.4 se presenta la clasificación de las KPAs del nivel 2 CMMI que son aplicables al modelo propuesto.

KPA	DESCRIPCION	ÁMBITO	APLICA
KPA8	RD Desarrollo de Requerimientos. (RequirementsDevelopment)	PS	SI
KPA9	TS Solución Técnica (TechnicalSolution)	-	NO
KPA10	PI Integración del Producto (ProductIntegration)	PS	SI
KPA11	VER Verificación (Verification)	PS	SI
KPA12	VAL Validación (Validation)	PS	SI
KPA13	OPF Enfoque en el Proceso Organizacional (OrganizationalProcessFocus)	O	SI
KPA14	OPD Definición del Proceso Organizacional (OrganizationalProcessDefinition)	O	SI
KPA15	OT Capacitación Organizacional.(Organizational Training)	O	SI
KPA16	RSKM	PS	SI

	Gestión de Riesgos 8 Risk Management).		
KPA17	DAR Análisis de Decisiones y Resoluciones (DecisionAnalysis and Resolution)	-	NO
KPA18	Ambiente de Desarrollo Integrado Producto y Proceso.(Integrated Product and Process Development (IPPD) nvironment	-	NO
KPA19	IPM GestiónIntegrada del Proyecto (Integrated Project Management)	PS	SI
KPA20	IT Integración de Equipo (IntegratedTeaming)	PS	SI
KPA21	ISM GestiónIntegrada de Proveedores (Integrated Supplier Management)	-	NO
KPA22	OEI Integración del Ambiente Organizacional (OrganizationalEnvironmentforIntegration)	O	SI

Tabla 2.4: Selección y clasificación KPAs aplicables NIVEL 3 CMMI
Elaborado por: M.Játiva, J. Carrera

2.2.1. KPAs NO APLICABLES NIVEL 3 CMMI

2.2.1.1. KPA 9: SOLUCIÓN TÉCNICA

El propósito del KPA de Solución Técnica es diseñar, desarrollar e implementar soluciones acorde a los requerimientos del producto. Esta KPA no se incluyó puesto que el modelo propuesto es considerado para empresas que cuenten con un departamento especializado en investigación de tecnología, el cual debe realizar los estudios necesarios para determinar la mejor solución tecnológica posible a ser utilizada en el desarrollo de un proyecto.

2.2.1.2. KPA 17: ANÁLISIS DE DECISIONES Y RESOLUCIONES

El propósito del KPA Análisis de Decisiones y Resoluciones es analizar posibles decisiones usando un proceso formal que evalúe las diferentes alternativas en base a sus criterios. Se comparan todas las alternativas y se

selecciona la mejor, sin embargo para este trabajo no se aplicará esta KPA, debido a que en metodologías ágiles el fin es probar si los requerimientos se van cumpliendo en los diferentes Sprints y de esta forma se va conformando el producto final lo que implica que a lo largo de todo el proyecto se tomará decisiones que se ajusten a lo requerido en ese momento.

2.2.1.3. KPA 18: AMBIENTE DE DESARROLLO INTEGRADO PRODUCTO Y PROCESO

Esta KPA se refiere al establecer un ambiente de trabajo en el que se involucre a las partes interesadas en el desarrollo del proyecto durante todo el ciclo de vida del mismo. De esta forma se asegura la satisfacción de todas las necesidades planteadas por el cliente.

Esta KPA no aplica al presente trabajo como una actividad aislada, puesto que dentro de SCRUM, cada Sprint considera la participación de cada parte interesada en el desarrollo del proyecto. La metodología ágil SCRUM adoptada genera necesariamente un ambiente de desarrollo integrado para su funcionamiento. La interacción de las partes involucradas se determina en las reuniones de planeación de proyecto, en las reuniones de planeación de Sprint y en el Scrum diario.

2.2.1.4. KPA 21: GESTIÓN INTEGRADA DE PROVEEDORES

En el Nivel 2 de madurez de CMMI no aplican las KP4 SAM referente a la Administración y Contratación de Proveedores puesto que el enfoque del modelo propuesto está en empresas en las que el negocio principal no es en sí el desarrollo de software, la administración y contratación de proveedores por tanto es realizada por una unidad específica y queda fuera de la competencia de la unidad encargada de los procesos de desarrollo de soluciones técnicas informáticas.

La KPA4 SAM se cubre también la gestión del software de subcontratación y para el caso de los proyectos abordados en este trabajo, tampoco es aplicable, puesto que todo el desarrollo debe ser realizado por el equipo de trabajo interno, tomando en cuenta especificaciones de requerimientos internos para equipos de hardware especializados que son utilizados en este tipo de proyectos.

Los requerimientos en el desarrollo de software son puntuales y se refieren a la utilización de equipos de hardware y específicos para manejo de datos, por esta razón no se requiere la subcontratación de software, cada desarrollo puede requerir de librerías o componentes especiales propias del equipo de hardware, estos paquetes son provistos por el fabricante y tanto el software como la asistencia técnica necesaria es contemplada dentro del contrato de adquisición de equipos y de mantenimiento.

2.3. MODELO DE GESTIÓN PROPUESTO MEDIANTE KPAs APLICABLES DE LOS NIVELES 2 Y 3 CMMI

En el presente trabajo se ha clasificado a las KPAs de nivel 2 y 3 de CMMI según sus características en dos grupos denominados ámbitos:

- **Ámbito Organizacional.**
- **Ámbito de Producto de Software.**

2.3.1. ÁMBITO ORGANIZACIONAL

Dentro de este ámbito se incluyen las KPAs cuyas actividades se encuentran directamente relacionadas con la estructura organizacional.

2.3.1.1. KPA 13: ENFOQUE EN EL PROCESO DE LA ORGANIZACIÓN

Esta KPA se refiere a la responsabilidad que tiene la organización en cuanto a la aplicación de los procesos en el desarrollo de software.

Se basa en los siguientes puntos:

1. Determinar las oportunidades de mejora del proceso.

- 1.1. Establecer las necesidades organizacionales del proceso.*
- 1.2. Evaluar los procesos de la organización.*
- 1.3. Identificar mejoras en los procesos de la organización.*

2. Planificar e implementar las actividades de mejora de los procesos.

- 2.1. Establecer los planes de acción para los procesos.*
- 2.2. Implementar los planes de acción para los procesos.*
- 2.3. Desplegar recursos organizacionales para el proceso.*
- 2.4. Incluir experiencias relacionadas con el proceso organizacional.*

Además se debe convocar cada 6 meses al equipo de desarrollo a revisar el proceso de desarrollo en base a las recomendaciones establecidas y de ser el caso revisar la modificación de procedimientos o herramientas utilizadas en el proceso de desarrollo de software para este tipo de proyectos.

2.3.1.2. KPA 14: DEFINICIÓN DEL PROCESO ORGANIZACIONAL

El objetivo de este proceso es establecer y mantener un conjunto utilizable de recursos organizacionales del proceso. Se desglosan las metas a conseguir con este proceso, y las prácticas que se requieren para conseguir estas metas: Los puntos que cubre son:

1. Establecer los recursos organizacionales del proceso.

- 1.1. Establecer procesos estándar.
- 1.2. Establecer descripciones del modelo de ciclo de vida.
- 1.3. Establecer criterios y líneas generales de adaptación.
- 1.4. Establecer un almacén de medidas de la organización.
- 1.5. Establecer la librería de recursos del proceso a nivel organizacional.

2.3.1.3. KPA 15: CAPACITACIÓN ORGANIZACIONAL

El propósito de este proceso es desarrollar las habilidades y conocimientos de las personas para que puedan desarrollar sus roles de forma eficiente.

Las actividades que cubre son:

1. *Habilitar a la organización para formar a su personal.*

1.1. Establecer las necesidades estratégicas de formación.

1.2. Determinar qué necesidades de formación son responsabilidad de la organización.

1.3. Establecer un plan táctico de formación para la organización.

1.4. Establecer la capacidad de formación.

2. *Proporcionar la formación necesaria.*

2.1. Dar la formación

2.2. Establecer registros de formación.

2.3. Determinar la efectividad de la formación.

2.3.1.4. KPA 22: INTEGRACIÓN DEL AMBIENTE ORGANIZACIONAL

El propósito del ambiente organizacional tiene como fin que el Producto, el Proceso de infraestructura de desarrollo, y la administración del personal sea integrado. Una infraestructura que maximiza la productividad y colaboración del personal se basa en un compromiso de toda la institución.

2.3.2. ÁMBITO PRODUCTO DE SOFTWARE

Dentro de este ámbito se incluyen las KPAs cuyas actividades se encuentran directamente relacionadas con el proceso de desarrollo de software en el Modelo propuesto.

Las actividades y procedimientos dentro del Ámbito de Producto de Software deben ser aplicados de manera repetitiva para cada uno de los SPRINT que se hayan definido en el proyecto.

2.3.2.1. KPA 1: GESTIÓN DE REQUERIMIENTOS

Esta KPA está cubierta con las siguientes actividades de SCRUM-XP:

- Gestión de la Pila del Producto o Backlog.
- Seguimiento de requerimientos en el Backlog
- Congelación de requerimientos durante cada sprint.
- Integración continua.
- Retroalimentación o interacción continua con el cliente.

2.3.2.1.1. GESTIÓN DE LA PILA DE PRODUCTO

Previo la gestión y la definición de la pila del producto, es importante establecer en forma clara los roles del proyecto SCRUM y la respectiva persona o personas que estarán desempeñando cada rol.

Los responsables por rol serán definidos utilizando el formato mostrado en la tabla 2.5

ROL DEL PROYECTO	NOMBRE RESPONSABLE	CARGO
Dueño del Producto	Representa la parte del cliente	Cargo que desempeña en la empresa.
Scrum Master	Facilitador del proyecto, asegura guía y quita obstáculos en la ejecución del proyecto. Se encarga de la gestión con el cliente para la definición de los requerimientos para cada sprint, hace de interfaz entre	Cargo que desempeña en la empresa.

		el cliente y el equipo.	
Equipo de desarrolladores		Responsables de crear el producto, formado por diseñadores, programadores y testadores. Estiman con el Scrum Master cada una de las tareas que formaran el Sprint.	Cargo que desempeña en la empresa.
Usuarios		Quienes utilizaran el software	Cargo que desempeña en la empresa.
Gerentes		Administradores de la organización	Cargo que desempeña en la empresa.

Tabla 2.5: Definición de roles del proyecto
Elaborado por: M. Játiva, J. Carrera

La primera actividad a realizarse en la gestión de la Pila del Producto o backlog, es la definición de la misma, esta es una parte fundamental del proyecto. La Pila del Producto constituye una lista de requisitos o funcionalidades a los que se ha asignado una prioridad. Es decir son elementos o funcionalidades que el cliente necesita del producto. Esta lista es escrita utilizando la terminología del cliente.

En este punto, la Pila de Producto es la vista de todo lo que podría ser hecho por el equipo en algún momento, en orden de prioridad.

Solo existe una única Pila de Producto; esto significa que el Dueño de Producto tiene que decidir sobre la priorización de todo el espectro.

Cada elemento de la Pila del Producto incluirán los siguientes campos:

- **ID:** Constituye un identificador único para cada elemento de la pila, es un secuencial.
- **Nombre:** Descripción corta del elemento de la pila. Esta descripción debe estar suficientemente clara como para que el cliente entienda de lo que se trata y se la pueda distinguir de los otros elementos.
- **Importancia:** Es el grado de importancia que el cliente da a este elemento, se debe definir la escala. Mientras más alto el valor, mas importante es el elemento, el valor de importancia, se define en el rango de 0 a 10.
- **Estimación Inicial:** Constituye la valoración inicial del equipo sobre la cantidad de trabajo necesario para la implementación de la historia. Esta estimación está realizada en “Días persona”, es decir en cuantos días trabajaría un equipo de desarrollo de dos personas en la implementación de esa historia sin distracciones durante 6 o 7 horas diarias, este es el número de horas productivas diarias que se definirá para el presente trabajo.
- **Como probarlo:** Es una descripción a alto nivel de cómo se probará esta historia en el producto final del Sprint. Es decir es esencialmente una simple descripción de cómo hacer una prueba o test para esta historia.
- **Notas:** Información o referencias que se necesite agregar a la historia, debe ser muy breve.

La estructura de cada elemento de pila se muestra en la tabla 2.6.

ID	Nombre de la Historia o elemento
Importancia	
Estimación Inicial	
Como probarlo	
Descripción	

Estado	
---------------	--

Tabla 2.6: Elemento de la pila del producto
Elaborado por: M.Játiva, J. Carrera

Estas tablas son almacenadas en un documento de Excel compartido en una unidad de red. El dueño del producto es oficialmente el propietario del documento, pero es necesario que este documento pueda ser accedido al resto de usuarios desarrolladores por razones de consulta o cambio de estimaciones especialmente.

Además para el caso del presente trabajo se establecerá un formato más general de la pila del producto en la que se podrá observar más fácilmente todos los elementos de la Pila del Producto.

El formato general de la Pila de Producto se muestra en la siguiente tabla 2.7.

ID	Elemento	Importancia	Estimación Inicial	Estado
1	Nombre del elemento de la Pila del Producto			
...	...			
...			

Tabla 2.7: Elementos de la pila del producto presentación general
Elaborado por: M. Játiva, J. Carrera

La tabla anterior muestra en conjunto de la lista de los elementos de la Pila de Producto. Esta tabla debe estar visible en forma impresa para todos los miembros del equipo y debe ser actualizada conforme el proyecto avance.

2.3.2.1.2. SEGUIMIENTO DE REQUERIMIENTOS EN LA PILA DE PRODUCTO (BACKLOG)

Es necesario asegurarse que Los requisitos que aparecen en la Pila del Producto sean negociables, evaluables, estimables y no demasiado grandes.

Además deben ser independientes puesto que el orden de implementación puede cambiar.

El seguimiento de los requerimientos en la Pila del Producto es parte de su gestión. En el presente trabajo, se entenderá por cambio en el backlog, a cualquier modificación que conlleve a la alteración del alcance del proyecto.

El Backlog pertenece al dueño del producto, y por tal razón este puede cambiar tantas veces como el dueño del producto crea necesario. Sin embargo en el Backlog inicial debe ser realizado de una forma de alto nivel, de tal manera que no debería sufrir cambios. No es recomendable el tener un Backlog que cambie constantemente.

Uno de los principales motivos de realizar un correcto seguimiento de los cambios en la Pila del Producto es el intentar que el dueño del producto no agregue cambios innecesarios.

Todos los cambios serán agregados al Backlog. Cada elemento de la pila contiene un atributo denominado "Estado", este campo o atributo tomará los valores de "original", "nuevo", "cambio", dependiendo de la acción realizada con el elemento del Backlog.

La actualización de los cambios en la Pila del Producto está a cargo del Dueño del Producto y debe ser discutido previamente con el Scrum Master. Cada cambio será comunicado al equipo en el SPRINT diario y además será reflejado y publicado en la tabla de Pila del Producto.

2.3.2.1.3. CONGELACIÓN DE REQUERIMIENTOS DURANTE CADA SPRINT

Un factor de importancia y uno de los pilares de SCRUM es que una vez que el Equipo se compromete en el desarrollo de un SPRINT, cualquier cambio o adición debe esperar al siguiente Sprint.

El presente trabajo acoge esto como una práctica inalterable, puesto que es necesario el que el equipo pueda trabajar con la certeza absoluta de que sus compromisos no cambiarán.

Si en el proceso de un SPRINT se determina que por alguna circunstancia de cambio de requerimientos, el trabajo del equipo para ese SPRINT constituirá una pérdida de tiempo, se atribuye al Dueño del producto o al equipo la capacidad de dar por terminado el SPRINT, con lo cual se deberá realizar una nueva reunión para la Planificación de un nuevo SPRINT deteniendo el cual es objeto de cambio.

2.3.2.1.4. INTEGRACIÓN CONTINUA

En el presente trabajo, se establecerá la integración continua como una práctica en la que cada miembro del equipo de trabajo debe integrar su trabajo al menos una vez al día, pudiendo haber múltiples integraciones diarias.

Cada integración debe ser verificada y validada por un BUILD especializado, de esta forma se detectarán y corregirán los errores de automatización el momento mismo de cada integración.

La integración continua es uno de los pilares para conseguir una consecución exitosa en el proyecto, asegurando que todo el sistema funciona en cada construcción o demo de manera coherente.

Para la consecución del proyecto se establecerá el uso de las herramientas informáticas de integración continua. Las características de la herramienta seleccionada en forma general deben ser:

- Mantener un repositorio de código.
- Automatizar la compilación e integración.
- Realizar pruebas.
- Commits diarios sobre el desarrollo.

- Cada commit debe compilar.
- Testar en clon de producción
- Resultados accesibles por parte de todos los miembros del equipo de desarrollo.

La integración debe ser realizada al menos una vez al día en la finalización de la jornada de trabajo de cada par de desarrolladores, esto no restringe a que en el transcurso del día se pueda realizar más integraciones si fuera necesario.

La herramienta a utilizar debe ser elegida en base al cumplimiento de las características anteriores y en función de la compatibilidad que la herramienta preste con respecto a la plataforma de desarrollo. En la siguiente tabla se muestra a manera de recomendación general una lista de herramientas que podrían ser utilizadas para la gestión de la integración de un proyecto.

2.3.2.1.5. RETROALIMENTACIÓN O INTERACCIÓN CONTINUA CON EL CLIENTE

La retroalimentación e interacción continua con el cliente se la obtiene durante la primera parte de cada planificación de cada Sprint, es necesario que el Dueño del Producto se encuentre presente en al menos en la primera de las reuniones de planificación de un Sprint. Este requerimiento es factible de implementar en el desarrollo de los tipos de proyectos que se quiere abordar en la implementación del presente trabajo, puesto que el correcto y oportuno funcionamiento del software constituye un factor determinante en la consecución exitosa de todo proyecto hacia el cual está enfocado , por esta razón es previsible con seguridad que el rol de Dueño de Producto se encuentre ocupado y cumpla con las actividades previstas, obteniendo así la retroalimentación e interacción con el cliente. En la sección correspondiente a la planificación del Sprint (KPA 2) se determina explícitamente las actividades que el rol Dueño de Producto debe cubrir.

2.3.2.2. KPA 2:PLANIFICACIÓN DE PROYECTOS DE SOFTWARE

Esta KPA está cubierta con las siguientes actividades de SCRUM-XP:

- Planificación SCRUM-XP al inicio de cada proyecto.
- Reuniones de planificación de cada Sprint al inicio de cada uno de estos.

2.3.2.2.1. PLANIFICACIÓN SCRUM-XP AL INICIO DE CADA PROYECTO

En realidad la planificación SCRUM-XP al inicio de cada proyecto constituye el primer paso a realizar, este paso consiste en la organización de la primera reunión del proyecto en el que el Dueño de Producto debe articular una visión detallada del proyecto y la funcionalidad requerida en el sistema informático a desarrollar para dicho proyecto.

Al final la reunión deberá evolucionar hacia una lista de requerimientos priorizada de funcionalidades del sistema informático y al entendimiento global sobre la planificación del proyecto, esta lista toma el nombre de la Pila de Producto.

El rol del Dueño de Producto es ocupado generalmente por el Jefe de Producción de la Planta de Artes Gráficas del IGM, puesto que este es el directamente responsable de la planificación y ejecución de dicho proyecto. En la reunión de planificación SCRUM-XP el Dueño de Producto debe exponer detalles de la planificación global del proyecto, puesto que la funcionalidad del sistema a desarrollar en gran medida dependerá de aspectos específicos de logística en cuanto a producción, aspectos como:

- Plazos límite para realizar la implantación e iniciar la producción del sistema informático o prototipo funcional.
- Logística de despacho de producto.
- Ubicación física de las terminales informáticas y módulos del sistema a utilizar en cada una de ellas.
- Necesidades en cuanto a tiempo de respuestas.
- Proceso de producción y puntos de control a implementar.
- Distribución del área destinada a la realización del proyecto.

Todos estos puntos son indispensables poner a conocimiento del equipo de desarrollo, puesto que en base a esto, se diseñará y seleccionará la mejor solución en cuanto a funcionalidad de software y utilización de dispositivos de hardware para afrontar desde la parte informática del proyecto.

Como entregables de esta reunión de planificación de SCRUM-XP se tendrá un documento denominado Acta de Planificación del Proyecto “NOMBRE DEL PROYECTO” y la Pila del Producto Inicial.

La estructura de la Pila del Producto se describe en la sección **de Gestión de la Pila de Producto (KPA1)**. El Acta de Planificación del Proyecto debe incluir los siguientes puntos generales como información mínima:

- **Encabezado:** Debe constar el nombre del proyecto, y la fecha en que se realiza la reunión.
- **Desarrollo:** Debe constar de las siguientes secciones:

- Sección de antecedentes: Describe información general sobre las razones o motivos producidos que llevan al planteamiento del desarrollo del proyecto.
- Desarrollo: En esta sección se debe detallar brevemente los acuerdos en funcionalidad y plazos a que se ha llegado durante la reunión y hacer referencia a la Pila de Producto, ésta última es también resultado de la reunión de Planificación SCRUM-XP al inicio del proyecto y además es aquí donde se establecen a detalle los requerimientos del sistema informático con sus respectivas prioridades.
- **Conclusiones y recomendaciones:** En esta sección se establece resoluciones tomadas en la reunión y se incluye también la descripción de los recursos necesarios que se debe involucrar en el desarrollo del proyecto. Este documento y específicamente esta sección debe ser utilizado para poner en conocimiento de la Dirección el planteamiento del proyecto y gestionar la aprobación y asignación de los recursos necesarios a utilizar en el desarrollo de la parte informática del proyecto de Artes Gráficas del IGM.
- **Pie de Acta:** Contiene las formas de responsabilidad del Dueño del Producto y del SCRUM Master.

El producto de esta reunión constituye el compromiso por parte del equipo de desarrollo y establece la Pila del Producto inicial.

2.3.2.2.2. REUNIONES DE PLANIFICACIÓN DE SPRINT AL INICIO DE CADA UNO DE ESTOS

Antes de iniciar con la primera reunión para la planificación del primer Sprint, hay que asegurarse que la Pila de Producto se encuentra terminada. En general se debe asegurar que se cumpla con los siguientes puntos:

- Existencia de la Pila de Producto.
- Todos los elementos de la Pila de Producto deben tener definidos valores de importancia, no es problema si los elementos menos importantes tienen definidos un mismo valor. Se debe tomar en cuenta que los valores de importancia únicamente se definen para obtener un ordenamiento por importancia de cada historia.
- El dueño debe comprender cada historia, normalmente él es el autor, pero pueden existir casos en los que otras personas añadan solicitudes, estas deben ser priorizadas finalmente por el dueño del producto.

Es importante que el Dueño del Producto o un representante de este asistan a esta reunión además del equipo de trabajo, esto es porque cada historia contiene tres variables muy dependientes unas de otras.

La figura 2.1 muestra las variables de cada elemento que forma parte de la Pila de Producto.



Figura 2.1: Variables del Elemento de la Pila de Producto
Elaborado por: M. Játiva, J. Carrera

El alcance y la importancia son fijadas por el Dueño del Producto, mientras que la estimación es proporcionada por el equipo. Estas tres variables deben ser discutidas durante la reunión y afinadas por medio de diálogo entre el equipo y el dueño del producto.

La reunión de planificación debe ser iniciada por el dueño del producto definiendo cuales son las historias más importantes, para posteriormente ser revisadas por el equipo para la asignación de la respectiva estimación, iniciando por la definida más importante. Es posible que durante la revisión de cada historia aparezcan dudas que al ser resueltas conduzca a redefinir la estimación para la historia.

Es importante que el dueño del Producto participe en esta reunión, esta participación no debe representar ningún problema para la empresa puesto que el cliente es interno y además se tiene claro la importancia del software con respecto al éxito global del proyecto.

La reunión debe cubrir y definir los siguientes tópicos:

- Definición de la Meta del Sprint.
- Definición de la lista de miembros del equipo de trabajo y nivel de dedicación.
- Inclusión de Historias en el Sprint (Pila de Sprint).
- Definición de la duración del Sprint.
- Definición de la hora y el sitio para el Scrum diario.
- Elaboración de Historias Técnicas.

2.3.2.2.1. DEFINICIÓN DE LA META DE SPRINT

Esta debe estar escrita en términos del negocio, es decir en términos en que la gente fuera del equipo pueda entender. La meta del Sprint tiene que ser algo que no se haya logrado aun. La meta del Sprint puede parecer tonta y superficial al inicio de la planificación del Sprint, pero es necesaria cuando el equipo comienza a sentirse confuso acerca del direccionamiento de su trabajo.

Para la documentación de la meta de un Sprint se hará uso del formato expuesto en la tabla 2.8 mostrada a continuación:

SPRINT No.	
Nombre SPRINT:	
Meta de SPRINT:	

Tabla 2.8: Definición de la Meta de Sprint
Elaborado por: M. Játiva, J. Carrera

2.3.2.2.2. DEFINICIÓN DE LA LISTA DE MIEMBROS DEL EQUIPO DE TRABAJO Y NIVEL DE DEDICACIÓN

En cada reunión de Sprint, el equipo se está comprometiendo seriamente a completar el trabajo en el tiempo que se defina, este compromiso necesita ser meditado para poder tener éxito.

Se debe determinar cuánto tiempo se puede comprometer cada miembro del equipo para el trabajo en el Sprint. Es importante que este tiempo sea estimado en forma efectiva en, para lo cual se tomará en cuenta lo siguiente:

- Día laboral medio, descontando el tiempo que pasarán en reuniones, leyendo el correo, comiendo, etc. Para la mayoría de la gente esto significa 5-7 horas de tiempo disponible al día para trabajo relacionado con el Sprint.
- La determinación del tiempo de trabajo de cada miembro del equipo debe ser escrito en una tabla como la que se muestra, esta tabla es elaborada para cada Sprint y constituye un compromiso a respetar por parte de cada miembro del equipo.

La tabla 2.9 muestra el formato a utilizar para la definición del equipo de desarrollo y su nivel de dedicación dentro del proyecto.

SPRINT No:	1
Nombre de SPRINT	“Nombre definido para le SPRINT”
Duración del Sprint	3 Semanas

Días laborables en el Sprint	15 días
-------------------------------------	---------

Nombre	Días disponibles durante Sprint	Horas Disponibles por día	Total horas disponibles
Santiago García	15	5	75
Carlos Zambrano	15	5	75
José Cerón	10	4	40

Tabla 2.9: Definición del Equipo y Nivel de Dedicación
Elaborado por: M. Játiva, J. Carrera

La definición de esta tabla debe realizarse para cada Sprint y debe ser aceptada por todos los miembros del equipo de trabajo, ya que constituye un compromiso aceptado, y de esto depende la conclusión exitosa del desarrollo del Sprint y del Proyecto.

Por las características de los proyectos hacia los cuales está enfocado el presente trabajo, es un requerimiento que los miembros del equipo de trabajo con rol de desarrollo sean asignados al proyecto a tiempo completo con una estimación de 6 horas productivas por día tomando en cuenta el tiempo requerido para el almuerzo y un periodo de tiempo para solución de imprevistos, que en este tipo de desarrollos se presentan por las características mismas de los proyectos que se abordan en el presente trabajo.

2.3.2.2.3. INCLUSIÓN DE HISTORIAS EN EL SPRINT (Pila de Sprint)

Esta actividad es una de las más importantes en la planificación de cada Sprint, la decisión de que historias de la Pila de Producto se debe incluir en la Pila de Sprint. El equipo de desarrollo toma la decisión de que y cuantas historias de la Pila de Producto serán incluidas en la Pila de Sprint y por tanto se comprometen a desarrollar durante su duración.

La decisión de que historias incluir en la Pila de Sprint debe ser realizada en base a las estimaciones iniciales asignadas a cada Historia en la Pila de Producto y el número de días hombre disponibles según el equipo de trabajo asignado y a la prioridad que se haya dado a cada elemento de la Pila del Producto, es decir se debe considerar las de mayor prioridad y tratar de no sobrepasar la duración del Sprint de dos semanas.

En esta sección también se debe desglosar a un nivel más detallado cada elemento de la Pila del Producto que se incluya en el Sprint, de tal forma que por cada elemento de la Pila, se tenga una o más actividades a realizar, cada actividad debe constar de su respectiva estimación inicial la cual debe ser coherente con la estimación inicial asignada al elemento de la Pila del Producto. Es decir la sumatoria de las estimaciones de cada actividad del elemento debe ser igual a la estimación inicial del Elemento de la Pila del Producto respectivo.

Para el desglose y asignación individual de estimación inicial de esfuerzo en cada actividad obtenida de cada elemento de la Pila de Producto incluida en el Sprint, se utilizará el formato mostrado en la tabla 2.10.

No. SPRINT:				
Nombre SPRINT:				
ID.	Nombre Elemento	Tarea de Sprint	Responsables	Est. I.
001	Búsqueda registro para captura de datos imagen	Búsqueda por número de cedula	García, Zambrano	
		Búsqueda por numero de comprobante	García, Zambrano	
		Búsqueda por apellidos	García, Zambrano	
002	Cierre diario de operación	Filtro por usuario y Centro de Movilización	Zambrano, Cerón	

			Zambrano, Cerón	
		Bloqueo de operación	Zambrano, Cerón	
			Total:	

Tabla 2.10: División de Historias para el Sprint Ilustración 1
Elaborado por: M. Játiva, J. Carrera

En la Tabla 2.10 se muestra las actividades a realizar en cada elemento de la Pila del Producto y además se debe incluir la estimación de esfuerzo por cada actividad.

2.3.2.2.2.4. DEFINICIÓN DE LA DURACIÓN DEL SPRINT

Uno de los resultados de la planificación del Sprint es la definición de su duración. Generalmente los dueños del producto prefieren los Sprints cortos, y los desarrolladores los Sprints largos, pero la duración del Sprint tiene que ser un valor de compromiso.

Para el caso del desarrollo de los proyectos técnicos informáticos abordados en este modelo propuesto, la duración del Sprint dependerá de la sumatoria de las estimaciones de duración de los elementos de la Pila del Producto incluidos en la Pila del Sprint considerando también la duración de cada Historia Técnica definida para el Sprint, pero se recomienda no definir Sprints de más de 2 semanas de duración, además este intervalo debe ser revisado y puesto a consideración dependiendo del plazo que el proyecto deba cumplir.

Dos semanas de duración para un Sprint es un tiempo prudencial para proporcionar agilidad corporativa y para recuperarse de los posibles problemas que aparezcan durante el Sprint.

2.3.2.2.2.5. DEFINICIÓN DEL SITIO Y HORA PARA EL SCRUM DIARIO

El primer Scrum diario es básicamente el lanzamiento, es donde todos deciden por dónde empezar el trabajo.

Se recomienda definir el Scrum diario en horas de la mañana y el sitio asignado debe ser una sala destinada para reuniones, donde el equipo se encuentre cómodo para efectuar la reunión.

2.3.2.2.6. ELABORACIÓN DE HISTORIAS TÉCNICAS

Son actividades que es necesario realizar, pero que no constituye entregables ni están directamente relacionadas con ninguna historia específica y no representan nada con un valor inmediato para el dueño del producto.

Las Historias técnicas se refieren por ejemplo a la instalación de un servidor de aplicaciones, instalación de un motor de base de datos específico, etc.

Cada Historia Técnica será documentada haciendo uso del formato mostrado en la tabla 2.11.

No. SPRINT:	
Nombre SPRINT:	
No.	TITULO DE LA HISTORIA TECNICA
Descripción: Descripción de las actividades de las que consta la Historia Técnica.	
Recursos: Lista de los recursos que se necesitarán para implementar las actividades descritas en la Historia Técnica.	
Tiempo Requerido: Estimación del Tiempo en horas necesarias para la implementación de la Historia Técnica.	

Tabla 2.11: Historia Técnica
Elaborado por: M. Játiva, J. Carrera

Cada uno de los ítems tratados como historias técnicas en la Reunión de planificación del Sprint deben ser plasmados utilizando el formato de la tabla 2.11, y constituyen un compromiso para los miembros del equipo que trabajará en ese Sprint.

2.3.2.3. KPA3: SEGUIMIENTO Y SUPERVISIÓN DE PROYECTOS DE SOFTWARE.

Esta KPA está cubierta mediante la presentación de pequeñas versiones incrementales resultado de cada Sprint, esto facilita la supervisión del proyecto e impide que este se desvíe de la planificación inicial.

La metodología Scrum abarca la supervisión de proyectos de software de dos maneras:

- Observaciones de primera mano mediante el Scrum diario.
- Gráficas del backlog, las cuales miden el avance conseguido al final de cada Sprint.

2.3.2.3.1. SCRUM DIARIO

El Scrum diario consiste en una reunión corta de máximo 15 minutos de duración, esta práctica es clave en la metodología Scrum y además cubre con el seguimiento y supervisión del desarrollo del proyecto.

Todo el equipo debe asistir a esta reunión. El objetivo de ésta reunión es la de informar a los demás miembros del equipo sobre el progreso y los obstáculos que se hayan suscitado hasta el momento.

En el Scrum Diario, cada miembro del equipo debe topar los siguientes puntos:

- Que han hecho desde la última reunión.
- Que tienen planificado hacer antes de la siguiente reunión.
- Cualquier bloqueo o impedimento que tengan.

Hay que tener en cuenta que el Scrum Diario no es una reunión de estado para informar a un jefe; es tiempo que dedica un equipo auto-organizado para compartir entre sus miembros lo que está pasando.

En caso de existir bloqueos o impedimentos, se deberá informar al Scrum Master, éste se responsabilizará en ayudar a los miembros del equipo a resolverlos.

Para el presente trabajo se recomienda que a estas reuniones no acudan al Scrum Diario ni jefes, ni gerentes ni otros puestos de autoridad percibida. Esto puede hacer sentir al equipo que está siendo “observado” –con la presión de informar de un gran progreso todos los días (una expectativa poco realista), y cohibe el informar sobre problemas, además tiende a socavar la autogestión del equipo.

En el Scrum diario se debe actualizar los avances de conseguidos en la Tabla de Sprint. La forma de actualización y manejo de la tabla se explica en el siguiente punto referido a Graficas del backlog.

Cada reunión de Scrum diario durante la duración del Sprint, debe ser registrado con la utilización del formato mostrado en la tabla 2.12:

SPRINT No.:			
Nombre de SPRINT: “Nombre del Sprint Respectivo”			
SCRUM No.	Fecha	Observaciones Generales	Recomendaciones

Tabla 2.12: Registro de Scrum Diario
Elaborado por: M. Játiva, J.Carrera

2.3.2.3.2. MEDICIÓN DEL AVANCE CON GRÁFICAS DE BACKCLOG

La actualización del avance debe ser diario, cada día se coloca la cantidad de trabajo restante necesario para finalizar cada actividad que se haya definido en el respectivo Sprint.

El seguimiento del avance debe ser realizado mediante la utilización de la Tabla de Sprint, esta tabla estará dispuesta y publicada en un lugar visible en el sitio que se ha definido para el Scrum diario (Sala de reuniones).

Lo importante de la actualización de la tabla y la gráfica del Backlog es que a parte del seguimiento al avance del desarrollo del proyecto, también se muestre al equipo el progreso hacia que se tiene hacia la consecución de su objetivo, este progreso no es mostrado en términos de cuanto se ha realizado hasta el momento, sino en término de cuanto trabajo queda en el futuro, es decir se debe mostrar en realidad la cantidad de trabajo que separa al equipo de conseguir su objetivo.

Si la línea de trabajo (en la gráfica del Backlog) restante no está bajando hasta la finalización del trabajo cuando se acerca el final del Sprint, entonces el equipo necesita hacer ajustes, por ejemplo reducir el alcance del trabajo o encontrar una forma de trabajar más eficientemente al tiempo que mantienen un ritmo sostenible.

Varios autores recomiendan el mostrar la Gráfica y la Tabla en papel en una pared, puesto que es más visible, rápida y está siempre en contacto con el equipo de trabajo, a diferencia de tenerla como una gráfica en un computador.

El formato de la Tabla de Sprint y su utilización se muestra a en la tabla 2.13.

SPRINT No:									
Nombre de Sprint:									
Esfuerzo estimado restante al final del día									
Elemento de la Pila del Producto	Tarea de Srpint	Responsables	Estimación Inicial	1	2	3	4	.	.
Búsqueda registro para captura de datos imagen	Búsqueda por número de cedula	García, Zambrano	10	9	9	6	5	.	.
	Búsqueda por numero de comprobante	García, Zambrano	10	9	8	6	5	.	.
	Búsqueda por apellidos	García, Zambrano	10	9	8	4	3	.	.
Cierre diario de operación	Filtro por usuario y Centro de Movilización	Zambrano, Cerón	8	8	7	3	2	.	.
		Zambrano, Cerón	9	8	7	5	4	.	.
	Bloqueo de operación	Zambrano, Cerón	10	9	9	2	1	.	.
		Total:	55	51	47	40	36		

Tabla 2.13: Tabla de Sprint
Elaborado por: M. Játiva, J.Carrera

Como se muestra en la tabla 2.13 de Sprint, el esfuerzo mostrado al final del día debe ser el esfuerzo faltante para terminar la actividad definida en el Sprint.

La utilización y formato de la Gráfica de Backlog se muestra en la figura 2.2.

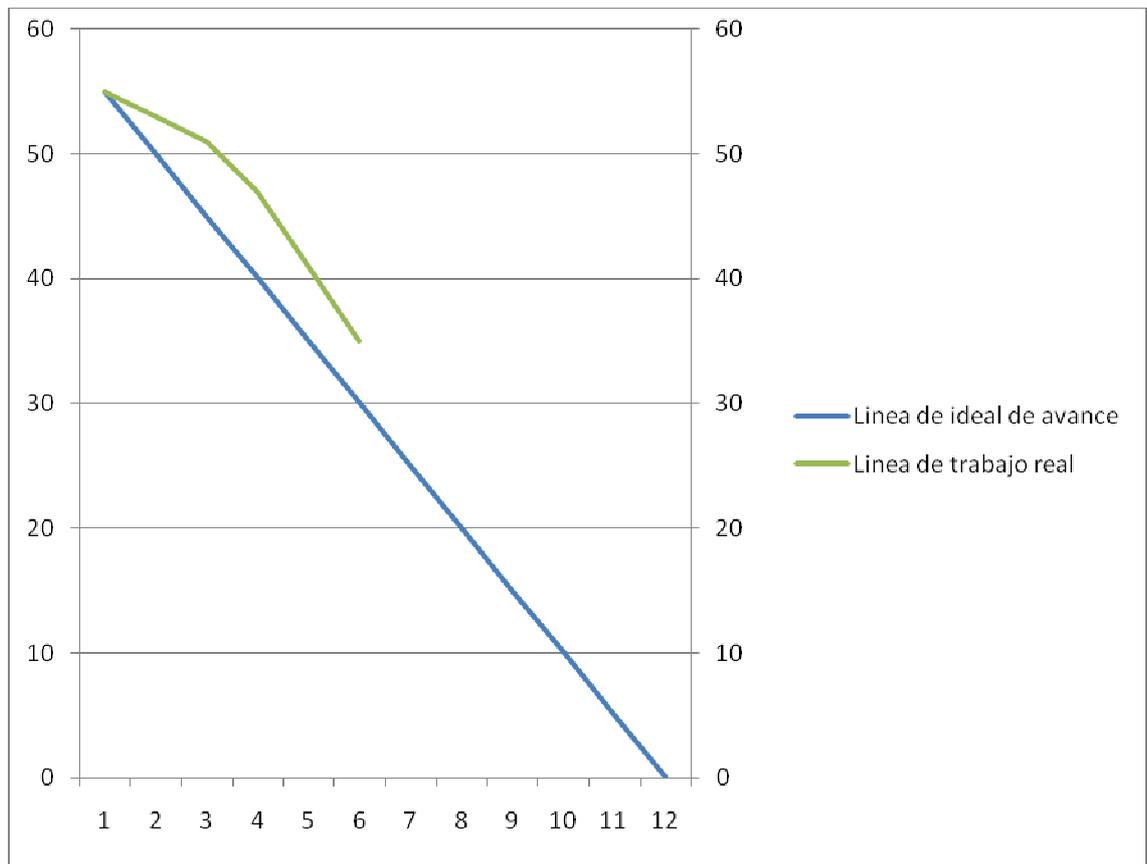


Figura 2.2: Gráfico de Backlog
Elaborado por: M. Játiva, J. Carrera

En la Gráfica de Backlog se puede apreciar de diferente forma la información de avance del equipo de desarrollo en el Sprint, en la gráfica se muestra el esfuerzo restante durante al finalizar cada día durante el desarrollo de un Sprint. En el eje de las X, se coloca los días estimados para el conjunto de elementos de la Pila de Producto que se hayan incluido en el Sprint y en el eje de las Y se coloca el esfuerzo aún requerido para cubrir completamente con el elemento de la Pila de Producto respectivo.

La Gráfica de Backlog muestra además el comportamiento del trabajo real con respecto al trabajo ideal, esto constituye una muy buena herramienta para realizar el seguimiento y supervisión ejecutiva durante el desarrollo del proyecto.

2.3.2.4. KPA 6: ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

En los proyectos desarrollados con metodologías ágiles y específicamente SCRUM-XP el aseguramiento de la calidad del producto mediante la implementación de una fase de pruebas, tiene un funcionamiento diferente a las metodologías tradicionales por las siguientes razones:

- No se cuenta con un equipo exclusivo dedicado a pruebas.
- En cada Sprint el equipo de desarrollo libera una aplicación potencialmente entregable, lo que implica que:
 - No se cuenta con suficiente tiempo para elaborar un plan de pruebas.
 - No es posible esperar hasta que el producto de cada Sprint se encuentre completamente libre de errores antes de iniciar con el nuevo Sprint.
 - No existe tiempo para realizar periodos de “retesting”.

En los proyectos implementados con metodologías ágiles no se puede establecer una fase de pruebas al final del proyecto. Las actividades de pruebas necesariamente deben ser introducidas dentro de cada iteración.

Esta KPA implica revisar y auditar los productos de software. En una organización CMM nivel 5, la empresa debe poseer un equipo definido e independiente que se dedica exclusivamente a estas actividades. El modelo se enfoca en empresas que no se encuentra en nivel 5 CMM y además sus actividades de negocio no se centran en el desarrollo de software, por esta razón se puede entender que las organizaciones a las que está dirigido el

modelo propuesto en este trabajo no cuenta con el recurso humano disponible para tareas propias y exclusivas de revisión y auditoría en los productos de software para el control de calidad del mismo.

El presente trabajo hace una abstracción al desarrollo de los proyectos Informáticos Técnicos de una empresa, tomando en cuenta la falta de disponibilidad de recursos suficientes que sean destinados y especializados a labores de control de calidad, se debe hacer uso de los cuatro recursos destinados a este tipo de proyectos que en este trabajo se propone, es decir cada par deberá realizar actividades de control de calidad y pruebas sobre el trabajo de desarrollo del otro par de desarrolladores.

Las pruebas posiblemente es el factor más variable entre estos tipos de proyectos, puesto que estas dependerán de factores como tipo de software que se encuentre desarrollando o el número de recursos asignados a estas actividades, que en cada proyecto es variable dependiendo de la disponibilidad.

Se debe considerar que el tiempo de desarrollo disponible en este tipo de proyectos es bastante limitado y generalmente no es posible extender ni un solo día el plazo de entrega del producto de software y su consiguiente puesta en producción, por estas razones es conveniente y recomendable que el control de calidad sea enfocado a pruebas de funcionalidad. Sin duda el desarrollo de software debe cumplir con estándares y el cumplimiento de estos debe ser verificado, sin embargo el tiempo es un factor determinante en la conclusión exitosa del proyecto.

El desarrollo en pares asegura que el cumplimiento de estándares de desarrollo sean cubiertos en el proceso mismo, lo cual disminuye el tiempo destinado a las pruebas en este sentido, enfocándose directamente a las pruebas de funcionalidad.

Las pruebas de software es una actividad en la cual el sistema o uno de sus componentes se ejecutan en circunstancias previamente definidas, los

resultados son observados y registrados, para posteriormente realizar una evaluación. El procedimiento de pruebas en el presente trabajo se establece de sobre:

1. Prueba de Funcionalidad.
2. Revisión de Presentación.
3. Revisión de Estándares de Codificación.

En la figura 2.3 se muestra las pruebas a aplicar sobre el producto de software resultante de cada iteración.



Figura 2.3: Pruebas de software
 Elaborado por: M. Játiva, J. Carrera

El equipo de pruebas debe ingresar y utilizar el sistema tal y como lo harían los usuarios finales, el equipo eventualmente encontrará errores, los mismos que serán reportados al equipo de desarrollo para su corrección.

El objetivo es asegurar la calidad del software producido, pero minimizando los recursos que se deban a signar para el periodo de pruebas. Esto se consigue maximizando la calidad en la generación de código, lo cual constituye ser la principal ventaja del trabajo en codificación en parejas.

Una práctica a considerar en el presente trabajo y que además constituye una forma de minimizar el trabajo en la fase de pruebas, es el considerar Sprints no

grandes, es decir no incluir demasiados elementos de la Pila del Producto en cada Sprint.

El procedimiento definido consiste en realizar la revisión del cumplimiento de estándares como funcionalidad del producto resultado de cada Sprint mediante una plantilla a modo de checklist.

2.3.2.4.1. PRUEBAS DE FUNCIONALIDAD

En la revisión de la funcionalidad se utilizará la plantilla siguiente, en función a la información de funcionalidad que consta para cada elemento que está formando parte de la Pila de Sprint, esta información es definida en la elaboración del a Pila del Producto.

La tabla 2.14 muestra el formato de la lista de chequeo a utilizar en el desarrollo de las pruebas para Revisión de Funcionalidad.

Revisión de Funcionalidad				
No. Sprint:				
Nombre SPRINT:				
Fecha:				
Actividad	Si	No	No aplica	Información adicional
¿El producto cubre con la funcionalidad especificada en la Pila de Sprint?				
¿Los datos de la forma cambian en forma sincronizada?				
¿El tiempo de respuesta es razonable para el usuario?				
Cuando se cambia el valor de un campo de entrada, ¿se modifica también el campo de despliegue?				
Los bloques hijos están coordinados con el bloque padre en consulta, borra doy cuando se limpia la forma?				

Los campos que hacen referencia a datos de tablas ¿tienen cada uno su lista de valores?				
¿Las listas de valores son lentas para recuperar la información?				
¿El orden de navegación de los campos es el correcto?				
¿Se ha implementado manejo de excepciones adecuadamente?				
¿Los campos Validate from LOV funcionan adecuadamente?				
¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?				
¿Si llama reportes, la extensión de los reportes es la correcta? (NO rdf, debe estar sin extensión).				
¿El desarrollo de la funcionalidad objeto de la revisión de este Sprint se encuentra debidamente integrado?				

Tabla 2.14: Revisión de Funcionalidad SPRITN
Elaborado por: M. Játiva, J. Carrera

2.3.2.4.2. REVISIÓN DE PRESENTACIÓN

La tabla 2.15 muestra el formato de la lista de chequeo a utilizar en el desarrollo de las pruebas para Revisión de Presentación.

Revisión de Presentación				
Sprint:		Descripción:		
Fecha:				
Analista:		Revisor:		
Actividad	Si	No	No aplica	Información adicional
¿Están claramente definidos los bloques de información (Frames)?				
¿Tiene los encabezados de título y nombre de aplicación correctos?				
¿Las etiquetas de los campos son claras y representativas?				
¿Los campos de despliegue están completamente inhabilitados y del color respectivo?				
¿Los campos de solamente despliegue están claramente				

identificados?				
¿Los campos fecha tienen el formato DD-MON-RRRR y se puede ingresar los datos como Ej: 12ago2001?				
Cuando se tiene una forma con múltiples tabs, ¿se conoce cuál es el registro padre de los tabs?				
¿Los Radio Groups tienen un frame que los abarca?				
¿Los campos están alineados en forma correcta?				
¿Los campos requieren y tienen Tooltip?				

Tabla 2.15: Revisión de Presentación SPRITN
Elaborado por: M. Játiva, J. Carrera

2.3.2.4.3. REVISIÓN DE ESTÁNDARES DE CODIFICACIÓN

La tabla 2.16 muestra el formato de la lista de chequeo a utilizar en el desarrollo de las pruebas para Revisión de Estándares de Codificación.

Revisión de Estándares de Codificación				
Sprint:		Descripción:		
Fecha:		Revisor:		
Analista:				
Actividad	Si	No	No aplica	Información adicional
¿Se ha hecho revisión por pares?				
¿Se ha realizado el proceso de afinamiento sql?				
¿Se encuentra el código debidamente documentado?				
¿Tienen las variables nombres fáciles de entender?				

Tabla 2.16: Revisión Estándares de Codificación SPRITN
Elaborado por: M. Játiva, J. Carrera

El período de Sprint debe contemplar tiempo para realizar la revisión del producto resultante previo su finalización.

Para el trabajo presente se recomienda no establecer Sprints largos, es decir no incluir demasiados elementos de la Pila de Producto, de esta forma se asegura que el tiempo que se tome para la corrección de errores que sean

encontrados en esta fase, tanto en las Revisiones de Funcionalidad, estándares de Presentación y Estándares de Codificación sean cortos, de tal forma que no altere el cronograma de ejecución del Sprint.

2.3.2.5. KPA 7: GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

La Gestión de la Configuración del Software es un proceso cuyo propósito es establecer y mantener la integridad de los productos de trabajo a través de los siguientes puntos:

- La identificación de los elementos/productos que van a ser controlados: Se debe establecer cuáles son los elementos de configuración.
- La definición de un procedimiento para el control de los productos: Establecer el cómo controlar los cambios sobre los elementos de configuración.
- El registro o informe del estado de los productos: Establecer el estado actual de los elementos de configuración.
- Las auditorías de configuración: Establecer si los elementos de configuración cumplen los requisitos.

Un elemento de configuración es cualquier producto de trabajo, tanto producto final como productos intermedios y tanto productos entregables al cliente como productos internos del proyecto, cuyo cambio pueda resultar crítico para el buen desarrollo del proyecto.

En la gestión de la configuración del software se hace uso del concepto de Línea Base, una línea base es un conjunto de elementos de configuración formalmente designados y fijados en un momento específico del ciclo de vida.

Los elementos incluidos en la línea base tendrán que cumplir unas condiciones mínimas, es decir, han de estar acabados y formalmente aprobados. La línea base sólo puede ser modificada a través de un procedimiento formal de cambios. La línea base, junto con todos los cambios aprobados sobre la misma, representa la configuración vigente y aprobada.

El principal objetivo de realizar gestión en la configuración del software es obtener integridad en el producto de software en cuanto a dos puntos específicos:

- Saber exactamente lo que se ha entregado al cliente
- Saber el estado y contenido de las líneas base y elementos de configuración.

Mediante la gestión de la configuración se busca el establecer una forma efectiva y eficiente de gestionar y comunicar los cambios en líneas base y elementos de configuración a lo largo del ciclo de vida.

Entre los problemas que se pueden generar al no realizar una correcta gestión de configuración se puede listar los siguientes:

- Entrega de una versión con errores.
- Entrega de una versión con cambios que no han sido probados
- Se puede encontrar errores en la versión entregada que no puede reproducirse en la última versión que el equipo de trabajo tiene.

Para el tipo de proyectos que se pretende abarcar en el presente trabajo se deberá realizar una abstracción en cuanto al cubrimiento de esta KPA, ya que el producto de software final no es distribuido comercialmente, el cliente es interno a la empresa y es el mismo equipo de desarrollo es el encargado de implantar y configurar todo lo necesario para el funcionamiento del software en el sitio designado.

Dadas las características de los proyectos abordados por el presente trabajo y las condiciones y restricciones en cuanto a la disponibilidad de recursos por parte de las empresas hacia las que se enfoca el modelo propuesto, no es posible centralizar el rol de Gestor de Configuración en un solo recurso. Las actividades de gestión de configuración recaerán sobre cada par de

desarrolladores que durante el desarrollo hayan tenido la necesidad de realizar cambios en el elemento.

Las prácticas generales a realizar son:

- Establecer líneas base.
- Establecer control de cambios.
- Establecer integridad.

Existen actividades que deben ser realizadas previamente que forman parte de la identificación de los elementos que serán incluidos en gestión de configuración. Estas actividades de identificación se muestran en la tabla 2.17.

Actividad	Descripción
Identificar los productos bajo gestión	Se debe identificar los elementos de software en los que es posible que durante el desarrollo del proyecto puedan estar sujetos a cambios.
Asignar atributos a cada elemento de configuración	Los atributos que se definirán para cada elemento son: <ul style="list-style-type: none"> • Identificador único. • Autor. • Tipo de archivo. • Responsable del elemento de configuración.
Definir estructura de almacenamiento	<ul style="list-style-type: none"> • Definir ubicación de almacenamiento. • Dentro de la carpeta correspondiente al elemento se coloca el elemento y un archivo que documente la línea base del elemento.
Definir un nivel de control	El equipo de desarrollo que tenga que hacer

de acceso de los miembros del equipo sobre la infraestructura de almacenamiento.	uso de un elemento, debe ser responsable del correcto manejo de las actualizaciones en el elemento y su línea base. El equipo de desarrollo tendrá acceso total.
--	--

Tabla 2.17: Gestión de Configuración de Software
Elaborado por: M. Játiva, J. Carrera

2.3.2.5.1. ESTABLECER LÍNEAS BASE

Una vez identificados los productos que estarán bajo gestión de configuración, se debe incluirlos bajo una línea base para que de esta forma a partir de ese momento, cualquier modificación que se vaya a realizar sobre dicho producto tenga que seguir los procedimientos diseñados para ello, es decir que cualquier cambio que sea aplicado sobre dicho producto sea correctamente registrado.

Para formar parte de la línea base no sólo tiene que estar identificado como elemento a incluir sino que tendrá que cumplir unas condiciones mínimas:

- El producto debe estar terminado.
- El producto debe estar probado.

La línea base del producto corresponde al producto finalizado y entregado para su integración en el sistema.

El evento que desencadena la incorporación a la línea base de un elemento es la realización de alguna tarea de aceptación formal del elemento.

Cada elemento de configuración estará identificado con el formato descrito en la tabla 2.18, esta tabla constituye la línea base del elemento.

No. Sprint:	
Nombre SPRINT:	
Nombre del Elemento:	
Id:	
Autor:	
Tipo de Archivo:	

Responsable del elemento:	
Almacenamiento:	
Descripción:	
Gestión de cambios	Agregar una descripción de cada cambio realizado sobre el elemento.
Acceso:	

Tabla 2.18: Línea Base
Elaborado por: M. Játiva, J. Carrera

Sobre esta tabla se debe reflejar la gestión de configuración sobre el elemento, esta tabla establece el seguimiento que se le hace a un elemento de configuración.

2.3.2.5.2. ESTABLECER CONTROL DE CAMBIOS

El primer paso para gestionar los cambios sobre los elementos controlados es determinar qué cambios realizar. El proceso de petición de cambios proporciona procedimientos formales para enviar y registrar peticiones de cambio, evaluar el coste e impacto potencial del cambio propuesto, y aceptar, modificar, o rechazar el cambio propuesto.

La petición de modificación y su aceptación, así como el análisis del impacto potencial de cambios sobre un elemento será realizado y discutido durante las reuniones de planificación de cada Sprint y durante el Scrum diario si hubiera consideraciones a tomar en cuenta durante el proceso de Sprint.

Tras realizar el cambio se comunicará, si así está establecido, a todos aquellos que estén afectados por dicho cambio. De esta forma, se pretende preservar la integridad.

Tras la comunicación del cambio, la correspondiente información en la línea base será actualizada, de esta actualización debe ser responsable el equipo de desarrollo que trabajó sobre un elemento.

2.3.2.5.3. ESTABLECER INTEGRIDAD

La integridad de los productos se manejará mediante la difusión de cualquier cambio durante las reuniones de Scrum diario y mediante la estructura de almacenamiento definido, de esta forma todo miembro del equipo de desarrollo del proyecto tendrá la información de la última versión del producto

2.3.2.6. KPA 8: DESARROLLO DE REQUERIMIENTOS

A medida que se vayan finalizando los Sprints se realice una retroalimentación en base a la tabla de elementos de la pila del producto y del registro de seguimiento de requerimientos (Backlog) realizados en el KPA2.

En el presente trabajo se fijará una fecha de retroalimentación al final de un número determinado de sprints en la que asista el ScrumOwner, Scrum Master y el equipo. Al final se aprobará la pila de producto inicial o se presentará la nueva.

El dueño del Producto ScrumOwner es quien finalmente abaliza el avance correcto del Proyecto, y quien puede pedir nuevos requerimientos que alteren la pila del producto (se debe aceptar este tipo de cambios en casos extremo), se debe trabajar conjuntamente con el equipo para determinar cualquier cambio y en forma conjunta establecer nuevas prioridades.

La tabla 2.19 muestra el formato a seguir en el establecimiento de un nuevo elemento de la Pila del Producto.

ID	Nombre de la Historia o elemento
Importancia	
Estimación	

Inicial	
Como probarlo	
Notas	
Estado	

Tabla 2.19: Nuevo Elemento de Pila de Producto
Elaborado por: M. Játiva, J. Carrera

Estos cambios también se deben ver reflejados en el burndown chart una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint

Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

2.3.2.7. KPA 10: INTEGRACION DEL PRODUCTO

El propósito es integrar el producto a partir de sus componentes, asegurar que el producto (como parte de la integración) funciona correctamente, y entregar el producto. Se debe cumplir con las siguientes prácticas específicas:

Las actividades que se realiza son:

1. Preparar la integración del producto:

- 1.1. *Determinar la secuencia de integración.*
- 1.2. *Establecer el entorno de integración del producto.*
- 1.3. *Establecer los criterios y procedimientos de integración del producto.*

2. Asegurar la compatibilidad de las interfaces:

- 2.1. *Revisar la completitud de las revisiones de las interfaces.*
- 2.2. *Administrar las interfaces.*

3. Integrar los componentes del producto y entregar el producto:

- 3.1. *Confirmar que los componentes del producto están listos para la integración.*
- 3.2. *Integrar los componentes del producto.*

3.3. *Evaluar las integraciones de los componentes del producto ya integrados.*

3.4. *Empaquetar y entregar el producto o componente.*

Cada una de estos pasos serán cubiertos con las actividades de SCRUM-XP descritas en la tabla 2.20.

Preparar Integración del Producto		SCRUM XP
	<i>Determinar la secuencia de integración.</i>	Acorde a la Pila de Productos y Prioridades
	<i>Establecer el entorno de integración del producto.</i>	Integración Continua
	<i>Establecer los criterios y procedimientos de integración del producto.</i>	Revisión de cada Sprint e Iteraciones

Asegurar la compatibilidad de las interfaces		SCRUM XP
	<i>Integrar los componentes del producto.</i>	Actualización Pila de Entrega
	<i>Revisar la completitud de las revisiones de las interfaces.</i>	Revisión de Sprint
	<i>Administrar las interfaces.</i>	Revisión de cada Sprint

Integrar los componentes del producto y entregar el producto		SCRUM XP
	<i>Confirmar que los componentes del producto están listos para la integración</i>	Refinamiento del Producto
	<i>Integrar los componentes del producto.</i>	Actualización Pila de Entrega
	<i>Evaluar las integraciones</i>	Revisión de Sprint
	<i>Empaquetar y entregar el producto o componente</i>	Sprints Entregados producto Final

Tabla 2.20: Integración de Producto
Elaborado por: M. Játiva, J. Carrera

Como se puede observar la aplicación de este KPA con el uso de Scrum-Xp está implícito en diferentes fases dadas en Scrum.

Las diferentes interacciones de cada Sprint tienen como finalidad llegar a la integración del producto. Durante el refinamiento inicial de la Pila de Producto y durante los refinamientos continuos de la pila de cada Sprint, el Equipo y el Dueño de Producto planificarán la entrega, perfeccionarán cada una de las características esperada del producto.

Para conseguir productividad en Scrum el equipo tiene que estar centrado en un producto o aplicación durante un Sprint.

La visión de Scrum es una visión de perfección en la que un producto es potencialmente entregable al final de cada Sprint, lo que implica que no se necesita trabajos de acabado tales como pruebas o documentación. Esto significa que todo está completamente terminado en cada Sprint; y que se podría realmente empaquetar o desplegar inmediatamente un producto después de la Revisión del Sprint.

Sin embargo debido a que en la práctica pueden suceder un sinnúmero de inconvenientes debido a que las instituciones en general manejan prácticas de desarrollo débiles y no se puede llegar a la visión de perfección, siempre quedará algo de trabajo por hacer, debido a esto se maneja el concepto de “Sprint de Entrega” lo cual viene a ser la integración del producto e incluye todas las tareas pendientes.

Si el equipo ha seguido buenas prácticas de desarrollo, con refactorización e integración continua, y pruebas efectivas durante cada Sprint, debería ser necesario poco trabajo para la KPA de Integración del producto.

Como entregable del producto integrado se deberá presentar un Demo en el que se pueda verificar que en efecto se cumple con todos los requerimientos solicitados y que la implementación es correcta.

En la tabla 2.21 se define el formato a utilizar en la finalización de un Sprint.

Demo	Sprint entregado
Sprint 1	Si/no

Tabla 2.21: Sprint entregado
Elaborado por: M. Játiva, J. Carrera

2.3.2.8. KPA 11: VERIFICACIÓN.

El propósito de esta KPA es el de asegurarse que el producto cumple o reúne los requerimientos establecidos.

La verificación constituye un proceso que se da a lo largo del desarrollo del proyecto, inicia con la verificación de los requerimientos, continua con la verificación de los productos generados a lo largo del trabajo y culmina con la verificación del producto terminado.

Las prácticas de VER permiten identificar defectos en etapas tempranas de la creación del producto y reducir los altos costos asociados a la identificación y corrección de defectos que se pueden presentar más adelante.

SCRUM como la mayoría de metodologías ágiles tiene el objeto de desarrollar productos funcionales en cada iteración, esto hace que las actividades que incluyen el cubrimiento de esta KPA sea incluido dentro de cada Sprint en la fase de control de calidad.

Para el presente trabajo se define a continuación las actividades que se incluyen en cada SPRINT para cubrir con esta KPA de CMMI:

Establecer la estrategia de verificación

Preparar el proceso de verificación.

- **Seleccionar productos de trabajo a verificar y métodos de verificación que serán usados:** Cada producto funcional obtenido al finalizar un Sprint se define como un elemento que será sometido a verificación.
- **Establecer y mantener el entorno necesario para dar soporte a la verificación:** El entorno que se utiliza para verificación es el establecido y utilizado a lo largo del desarrollo del producto, es decir el ambiente de desarrollo reproducido de forma similar al ambiente que se tendrá cuando el producto final se encuentre en producción.
- **Establecer y mantener procedimientos y criterios de verificación para los productos de trabajo seleccionados:** Cada requerimiento funcional definido para el Sprint constituye en criterio de verificación, para este trabajo se define el formato a utilizar en la tabla 2.22.

VERIFICACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint:			Fecha:		
Verificación No.:					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones

Tabla 2.22: Verificación
Elaborado por: M. Játiva, J. Carrera

El procedimiento es considerar y proporcionar las entradas reales de prueba en el producto y verificar las salidas producidas, se realizara observaciones y correcciones en caso de no cumplir con las salidas esperadas.

Esta plantilla debe ser construida al iniciar el Sprint, pero puede ser modificada durante el transcurso del mismo si fuera necesario.

Ejecutar las revisiones entre pares

Ejecutar las revisiones entre pares sobre los productos de trabajo seleccionados

- **Preparar las revisiones entre pares de productos de trabajo seleccionados:** El presente trabajo adopta la programación en pares, la ejecución de la verificación estará a cargo de cada par de desarrolladores en su producto de Sprint.
- **Llevar a cabo las revisiones entre pares sobre productos de trabajo seleccionados e identificar problemas resultantes de las revisiones:** Se ejecuta la verificación del producto con la utilización de la tabla definida.
- **Analizar datos sobre preparación, realización y resultados de las revisiones entre pares:** El proceso de verificación arroja resultados de cumplimiento que serán registrados en la tabla definida, así como las observaciones pertinentes, estos datos deben ser considerados por el par de desarrolladores para realizar las correcciones necesarias sobre el producto de Sprint antes de volver a ejecutar un nuevo proceso de verificación.

Ejecutar la verificaciones

Los productos de trabajo seleccionados son verificados contra los requerimientos especificados.

- **Realizar la verificación sobre productos de trabajo seleccionados:** Para el trabajo presente, esta actividad es cubierta con la verificación en pares, actividad que debe ser repetida hasta observar los resultados esperados, Esta decisión es tomada puesto que tratándose de una metodología de desarrollo como SCRUM que constituye ser iterativa en

interactiva con el, y dado las características de los proyectos abordados, se opta por cubrir esta actividad con la verificación en pares.

- **Analizar resultados de todas las actividades de verificación:** Los resultados producto de la revisión por pares son analizados y las correcciones son realizadas en base a estos antes de dar por finalizado el correspondiente Sprint.

Como parte del proceso para el cubrimiento de esta KPA, se llevara un registro de verificación para efectos de retro alimentación, este proceso debe ser repetido como parte del Sprint hasta conseguir los resultados deseados.

2.3.2.9. KPA 12: VALIDACIÓN

El objetivo de esta KPA es demostrar que un producto está apto a ser usado por parte del usuario final en el ambiente de producción.

Esta KPA es parecida a VER, con un enfoque diferente, aquí el producto a verificar es aquel que va a ser utilizado por el usuario final.

Las actividades para el cumplimiento de esta KPA son incluidas dentro de VER, puesto que como resultado de cada SPRINT se obtiene un producto funcional, que debe estar listo a ser utilizado por el usuario final.

2.3.2.10. KPA 16 GESTIÓN DE RIESGOS

La aplicación de Gestión de Riesgos en metodologías tradicionales es realizada como un proceso paralelo que incluye su propia agenda de tareas. Debido a que las metodologías tradicionales utilizan modelos con ciclos de vida en cascada donde cada proceso se realiza uno a continuación de otro y se planifica todo a su mínimo detalle, la gestión de riesgos permite controlar variables que podrían no permitir la finalización exitosa del producto en forma

explícita, sin embargo con el uso de metodologías ágiles como se propone en este trabajo el panorama cambia y la aplicación de Gestión de Riesgos se la hace de forma implícita en la propia metodología de Scrum como una disciplina articulada de manera natural en todas las actividades que se llevan a cabo en el proyecto, es decir se lo realiza de manera integrada en el propio proceso y, por lo tanto, de manera continua.

Existen diferentes criterios en relación a quién debe gestionar el riesgo en Scrum, por un lado se sugiere que es responsabilidad del dueño del producto gestionar el riesgo. El ScrumMaster y el equipo debería ayudar al dueño del producto a la óptima priorización del backlog. Otros criterios sugieren que la gestión del riesgo es una responsabilidad del equipo y todos en el equipo de Scrum tiene que trabajar en forma colaborativa para resolverlos.

La gestión de riesgos se realizará en las siguientes fases:

Scrum diario en las revisiones de sprint, donde se puede detectar riesgos los mismos que pueden ser presentados y resueltos.

Impediment Backlog.- Sería la parte más explícita de la gestión de riesgos la cual es una lista de incidencias que tienen que ser resueltas por el equipo y que el Scrum Master debe gestionar y asignar a alguien para que trabaje en ellas, y que será revisada en la reunión SCRUM diaria, utilizando preguntas como: ¿Qué está bloqueando el progreso del proyecto? O ¿Qué impedimentos se ha encontrado?. Responder estas preguntas día a día es sin duda una manera implícita de gestionar los riesgos del proyecto.

Sprint Retrospective, que permite atajar todos los riesgos relacionados con la comunicación con el cliente y los requisitos. El Scrum Master debe hacer hincapié en no solo hablar de impedimentos actuales sino de también de aquellos impedimentos que se vislumbran en el futuro del proyecto. Esto cubre perfectamente la detección de riesgos, pero ¿Qué hacemos para mitigarlos?.

Los riesgos que se tomaran en cuenta se muestran en la tabla 2.23:

Riesgo	Cómo lo controla Scrum
Ampliación descontrolada de características	En Scrum se parte de una lista ordenada por retorno de la inversión de las características del proyecto. Esta lista es gestionada por el Product Owner y puede ser establecida al principio del proyecto. Aunque siempre está abierta a la inclusión de nuevas características, esto siempre implica, de manera explícita un aumento en el número de Sprint, en la magnitud del proyecto. Esto ataja los problemas habitualmente relacionados con la ampliación descontrolada de características, pues en Scrum nunca la ampliación de características es descontrolada sino el fruto de un proceso explícito de priorización y evaluación.
Captura de requisitos realizada mal	Ninguna metodología protege de una captura de requisitos mal realizada. Pero una metodología si puede proteger los riesgos más habituales en lo relativo a la captura de requisitos: no realizarla en absoluto, realizarla de una manera demasiado exhaustiva, no asumir que cambiarán y no detallarlos suficientemente antes de la implementación. En Scrum estos riesgos se atajan respectivamente con la construcción del productbacklog, evitando detallar los requisitos en una fase temprana de proyecto y detallarles adecuadamente, durante el Sprint Planning Meeting, cuando se va a realizar su implementación.
Calidad insuficiente	La imposición que Scrum realiza de demostrar de la funcionalidad implementada durante el Sprint Review actúa como fusible de seguridad ante la calidad insuficiente. Ningún equipo que desarrolle software de baja calidad podrá salir airoso de un Sprint Review.
Plazos optimistas impuestos	En Scrum el equipo es el último responsable de aceptar los plazos y de comprometerse con la cantidad de características a implementar durante el Sprint. Nadie puede imponer plazos que no sean realistas pues el equipo tiene la potestad para no aceptarlos, con lo que los riesgos relacionados con la imposición de plazos optimistas no realistas se zanja de raíz.
Diseño inadecuado	De nuevo el Sprint Review y la demostración que realizamos durante el mismo actúa aquí como una válvula de escape que nos alerta rápidamente de las carencias en lo relativo al diseño inadecuado. Si el diseño de la aplicación no es adecuado, las carencias se harán patentes durante el Sprint Review. Durante la Sprint Retrospective también el equipo detecta a menudo partes del diseño de la aplicación que deben ser refactorizadas.
Desarrollo orientado a la investigación	El concepto de 'Hecho' es clave en Scrum. Solo se demuestran las características que están completadas, por lo tanto si investigamos y no logramos resultados la situación se hace patente de manera clara y rápida al no verse avances en el Product Burndown Chart y durante los Sprint Review. Scrum tiene sitio para el desarrollo orientado a la investigación pero siempre deja patente si esta investigación obtiene resultados o no.
Personal inadecuado	Ninguna metodología actúa de manera explícita en este aspecto. Scrum minimiza las posibilidades de que en el equipo haya personal inadecuado al incidir mucho en que los miembros del equipo sean multidisciplinares y que se produzca una continua comunicación y transmisión de conocimientos entre ellos mediante el Daily Scrum.

	De cualquier modo este es un problema más relacionado
Fricción con los clientes	Scrum como metodología ágil que sigue el manifiesto ágil y por tanto el principio de poner 'la colaboración con el cliente sobre la negociación de contratos'. Para ello Scrum pone en todo proyecto un representante de los intereses de cliente, el Product Owner y además permite y persigue la colaboración y la comunicación con todos los involucrados en el proyecto durante los Sprint Reviews.

Tabla 2.23: Gestión de Riesgos
Elaborado por: M. Játiva, J. Carrera

2.3.2.11. KPA 19: GESTIÓN INTEGRADA DEL PROYECTO

En XP y Scrum las actividades de gestión de la configuración son administradas usando el concepto de integración continua, donde se destina un computador únicamente para la tarea de integración.

Las características adicionales para la definición de la Herramienta de Integración:

- Características de la herramienta software como:
 - Integración de sistema de control de versiones.
 - Capacidad de manejo e integración de plataformas de construcción y despliegue.
 - Metodología de feedback e informes.
 - Etiquetado.
 - Si soporta dependencia de otros proyectos.
 - Facilidad de extender.
- Fiabilidad y rendimiento, hacer pruebas de rendimiento fiables.
- Sin perder la funcionalidad y robustez de una herramienta consolidada y establecida, hay que tener una visión de futuro, con la posibilidad de extender el producto y madurarlo en el futuro.
- Entorno objetivo. Hay que tener también una visión más allá de la calidad del código y no perder de vista el entorno de producción, trabajo y ejecución. Se debe tener en cuenta los sistemas operativos soportados

por los servidores, los requerimientos del sistema, el versionado y el soporte de versionado.

- Facilidad de uso. Aunque a veces este criterio puede ser subjetivo, el uso de consolas web, configuración a través de edición de archivos, utilización de asistentes o paneles de control vendrán determinados por los usuarios de la herramienta-servidor que va a ser utilizada y administrada.

Existen en el entorno varias herramientas de software, las cuales pueden ser utilizadas facilitar la cobertura del requerimiento que plantea esta KPA, las características generales de estas herramientas se resumen a lo siguiente:

- Mantener un repositorio de código.
- Automatizar la compilación e integración.
- Realizar pruebas.
- Commits diarios sobre el desarrollo.
- Cada commit debe compilar.
- Testar en clon de producción
- Resultados accesibles por parte de todos los miembros del equipo de desarrollo.

En la sección de anexos se muestra a manera de recomendación general una lista de herramientas que actualmente se encuentran en el mercado y que podrían ser utilizadas en el desarrollo proyectos.

2.3.2.12. KPA 20: INTEGRACIÓN DEL EQUIPO O COORDINACIÓN INTERGRUPAL

El concepto de integración del equipo tiene que ver con que todos y cada uno de sus miembros tenga la capacidad de resolver cualquier problema técnico o de requerimientos relacionados con el proyecto.

Esta KPA es cubierta por metodologías XP Y Scrum. La metodología SCRUM se enfoca en las prácticas de organización y gestión, mientras que las practicas XP son enfocadas a las prácticas de programación, es por esta razón que las dos metodologías se complementan bien.

En Scrum el equipo es “multi-funcional”, tiene todas las competencias y habilidades necesarias para entregar un producto potencialmente distribuible en cada Sprint. Es “autoorganizado” (auto-gestionado), con un alto grado de autonomía y responsabilidad.

En Scrum, los equipos se auto-organizan en vez de ser dirigidos por un jefe de equipo o jefe de proyecto. El equipo decide a que se compromete, y como hacer lo mejor para cumplir con lo comprometido; en el mundo de Scrum, al equipo se le conoce como “Cerdos” y a todos los demás como “Gallinas” (que viene de un chiste sobre un cerdo y una gallina que están hablando sobre abrir un restaurante llamado “Huevos con jamón”, y el cerdo no lo ve claro porque “él estaría verdaderamente comprometido, pero la gallina solo estaría implicada”). Para el caso de que cada miembro del equipo tenga un vasto conocimiento técnico se aplica la metodología XP con la programación en parejas

Se designará equipos de dos personas para la codificación, una persona tomará el nombre de “piloto” y es quien llevara manejará el teclado del computador donde se está realizando el desarrollo, la segunda persona toma el nombre de “navegante” y también debe poseer un computador en el cual se realizará tareas pequeñas cuando sea necesario, consultar documentación, etc.

La programación en parejas debe mejorar la calidad de código y mejorar la concentración del equipo. Esta forma de programar para el caso de los tipos de proyectos abordados en el presente estudio debe ser realizada tomando en cuenta los siguientes puntos:

- La programación en parejas no debe hacerse durante todo el día, o se debe tomar intervalos de descanso de 10 minutos cada 2 horas, puesto que resulta extremadamente cansado.
- Si bien el desarrollo se realiza en un solo computador, cada miembro de la pareja debe poseer su propio PC.
- Cambio de parejas frecuentemente, pero al inicio de cada Sprint.
- No forzar a ningún desarrollador a realizar programación por parejas, en ocasiones se puede perder un buen desarrollador por intentar obligar a algún recurso a inmiscuirse en esta forma de codificar. En estos casos se debe proporcionar las herramientas adecuadas y motivarlas en cuanto a la programación en parejas, pero permitirles experimentar a su propio ritmo.

CAPÍTULO 3 : APLICACIÓN A UN CASO DE ESTUDIO

Este capítulo estará centrado en la aplicación en un caso de estudio para efectos de validación del Modelo generado y que se encuentra documentado en el Capítulo II del presente trabajo.

3.1. SELECCIÓN DEL CASO DE ESTUDIO

El caso de estudio consiste en seleccionar un proyecto real que haya sido asignado a la Gestión Artes Gráficas, este proyecto debe contener la parte de desarrollo informático con las características descritas en el Capítulo I.

El proyecto seleccionado para la validación del Modelo de Gestión de Calidad para Desarrollo de Proyectos Informáticos de la Gestión Artes Gráficas del IGM es el PROYECTO DE EMISION DE TARJETAS DE IDENTIFICACIÓN DE LAS FUERZAS ARMADAS DEL ECUADOR, proyecto que se lo conocerá como DIRMOV, en relación a las siglas de la entidad encargada de la emisión de dichos documentos de identificación, Dirección de Movilización del Comando Conjunto de las Fuerzas Armadas.

3.1.1. DESCRIPCIÓN DEL CASO DE ESTUDIO SELECCIONADO

Siendo el Instituto Geográfico Militar del Ecuador una entidad estatal que posee tecnología e infraestructura para el suministro de soluciones gráficas y de seguridad documentaria, y siendo el IGM una empresa pública, se hace la adjudicación del Proyecto para la Emisión de Tarjetas de Identificación para Las Fuerzas Armadas del Ecuador, DIRMOV.

Este proyecto DIRMOV consiste en la emisión y personalización de 485000 tarjetas de identificación Militar a nivel Nacional, distribuido en 39 centros de emisión fija a nivel nacional y 4 equipos móviles.

El proyecto para la Unidad de Producción de la Gestión Artes Gráficas consiste en la elaboración de las especies según los requerimientos de seguridad documentaria establecidos y la posterior personalización de cada tarjeta de identificación en los Centros de Movilización ubicados a lo largo del país.

Los requerimientos de seguridad documentaria ni los aspectos de propios de producción de la especie constituyen parte del Proyecto Técnico Informático, así que para el presente trabajo se obviará mencionados aspectos.

La parte concerniente al proyecto informático y en la cual el Modelo de Gestión será probado es el sistema de emisión que el IGM debe proveer como una parte del Proyecto DIRMOV. El IGM debe proveer una solución informática de software que garantice la correcta personalización y emisión de los Documentos de Identificación Militar en los centros y equipos móviles mencionados anteriormente.

El tiempo asignado antes de realizar la primera emisión es de 60 días calendario y no se puede extender el plazo por ningún motivo, además la aplicación de software debe ser implementada bajo plataforma Microsoft Visual Estudio 2008 compatible con la versión 2010, estructura cliente servidor (stand alone), el repositorio de datos se encuentra bajo un motor de base de datos Sybase.

La aplicación deberá ser capaz de conectarse a la base de datos centralizada perteneciente a la DIRMOV y ser capaz de emitir la Tarjeta de Identificación respectiva con la personalización correspondiente.

A continuación se afrontará el proyecto seleccionado DIRMOV con el Modelo de Gestión desarrollado en el Capítulo II, cabe resaltar que la aplicación informática como se puede notar constituye una parte fundamental dentro del proyecto DIRMOV, el fracaso en la implementación del Proyecto Informático

constituiría el fracaso en el proyecto en conjunto de la Gestión Artes Gráficas del IGM.

3.2. IMPLEMENTACIÓN DEL CASO DE ESTUDIO SELECCIONADO UTILIZANDO EL MODELO DE GESTIÓN DESARROLLADO.

3.2.1. ÁMBITO ORGANIZACIONAL

3.2.1.1. KPA 13: ENFOQUE EN EL PROCESO DE LA ORGANIZACIÓN

Esta KPA se refiere a la responsabilidad que tiene la organización en cuanto a mejorar los procesos en el desarrollo de software basándose en el conocimiento de las fortalezas y debilidades actuales de los procesos ya establecidos. Para el caso del presente trabajo, ésta responsabilidad recae a nivel de la Gestión Tecnológica del IGM.

La Gestión Tecnológica debe garantizar que se planifican adecuadamente actividades para obtener la mejora de los procesos de desarrollo de software.

El marco de trabajo Scrum permitirá revelar debilidades en los procesos rápidamente y proporcionará caminos para que el personal de Tecnología explore nuevas formas de resolver los problemas de procesos en ciclos cortos y con pequeños experimentos de mejora.

Las oportunidades de mejora deberán ser identificadas al final de cada Sprint (Post Morten), estas serán recomendadas a nivel de Coordinación de Gestión Tecnológica para ser aplicadas en proyectos posteriores y elevados a conocimiento de la gerencia del IGM en las habituales reuniones de Staff realizadas los días lunes.

En el plan estratégico del IGM se tiene claramente definido los objetivos de la Institución, las acciones para alcanzar estos objetivos están descritas en un conjunto de procesos existentes los cuales en parte se han definido en base a las debilidades existentes, entre ellos consta el proceso de: Optimizar la

Gestión de Tecnología, el mismo que puede apoyarse con los lineamientos de Scrum y hacerlos más específicos y realizables.

En la tabla 3.1 se muestra en enfoque en los procesos de la organización definidos en el Plan Estratégico del IGM.

4.	MAXIMIZAR LA CAPACIDAD DE GESTION HUMANA Y TECNOLOGIA	FACTOR	+	POND	30
4.1	OPTIMIZAR LA GESTION HUMANA	FACTOR	+	POND	50
4.1.1	FORTALECER LA GESTION DEL TALENTO HUMANO Y LA SEGURIDAD INDUSTRIAL	FACTOR	+	POND	50
4.1.2	MEJORAR LA COMUNICACION INTERNA	FACTOR	+	POND	30
4.1.3	MEJORAR EL CLIMA LABORAL	RESULTADO DE LA ENCUESTA DE CLIMA LABORAL	+	PROM	20
4.2	OPTIMIZAR LA GESTION TECNOLOGICA	FACTOR	+	POND	50
4.2.1	AMPLIACIÓN DE LOS SERVICIOS DEL INTRANET	% CUMPLIMIENTO DE PLAN	+	PROM	20
4.2.2	MANTENER LA INF. CRITICA RESPALDADA	% DE INFO RESPALDADA	+	PROM	25
4.2.3	CALIDAD DEL SERVICIO PARA EL CLIENTE INTERNO	#CASOS RESUELTOS/ #CASOS PEDIDOS	+	PROM	15
4.2.4	MANTENER RED DE COMUNICACIÓN DISPONIBLE.	% TIME UP DE COMUNICACIÓN	+	PROM	15
4.2.5	IMPLEMENTAR UNA RED INALAMBRICA.PARA INTERNET	% CUMPLIMIENTO DE PLAN	+	PROM	5
4.2.6	APLICAR PLAN DE LICENCIAMIENTO DEL SOFTWARE.	% CUMPLIMIENTO DE PLAN	+	PROM	10
4.2.7	RENOVACIÓN DE EQUIPOS TECNOLÓGICOS	CUMPLIR CON EL PLAN DE RENOVACIÓN.	+	PROM	10

Tabla 3.1: Enfoque en Proceso de la Organización

Elaborado por: M. Játiva, J. Carrera

3.2.1.2. KPA 14: DEFINICION DEL PROCESO ORGANIZACIONAL

A continuación se describe como se encuentra definido el Proceso Organizacional del IGM, el mismo que fue implantada en el año 2007 y se encuentra vigente hasta la fecha.

En la figura 3.1 se muestra el mapa de procesos definidos en el Plan Estratégico del Instituto Geográfico Militar.



Figura 3.1: Mapa de Procesos IGM
Elaborado por: M. Játiva, J. Carrera

3.2.1.2.1. MAPA DE PROCESOS DEL INSTITUTO GEOGRÁFICO MILITAR

Es la representación gráfica de la interacción de los procesos que intervienen en la gestión institucional del Instituto Geográfico Militar, responsable de la generación de productos y servicios que demandan los clientes/usuarios internos y externos.

3.2.1.2.1.1. ESTRUCTURA POR PROCESOS IGM

Procesos del Instituto Geográfico Militar.- Los procesos que elaboran los productos y servicios del Instituto Geográfico Militar, se ordenan y clasifican en

función de su grado de contribución o valor agregado al cumplimiento de la misión institucional.

Los procesos gobernantes orientan la gestión institucional a través de la formulación de políticas y la expedición de normas e instrumentos para poner en funcionamiento a la organización.

Los procesos que agregan valor generan, administran y controlan los productos y servicios destinados a usuarios externos y permiten cumplir con la misión institucional.

Los procesos habilitantes están encaminados a generar productos y servicios para los procesos gobernantes, agregadores de valor y para sí mismos, viabilizando la gestión institucional.

Dentro de la toda la estructura de procesos del IGM se detallará solamente el área de Gestión Tecnológica y particularmente el Proceso de Desarrollo de Aplicaciones.

3.2.1.2.1.1 GESTIÓN TECNOLÓGICA

Proceso de: Planificación Informática

Proceso de: Desarrollo de Aplicaciones

Proceso de: Mantenimiento de Hardware y Software

Proceso de: Administración de Infraestructura Tecnológica

El proceso de Desarrollo de Aplicaciones está descrito en base a procedimientos tradicionales para su ejecución. Es tema de este trabajo proponer nuevos lineamientos a ser aplicados en el desarrollo de aplicaciones de la Unidad de Artes Gráficas que realizan proyectos netamente técnicos en relación a aplicaciones administrativas desarrolladas en otras unidades. Sin embargo la definición del proceso organizacional se mantiene.

3.2.1.3. KPA 15: CAPACITACION ORGANIZACIONAL

En el IGM se realiza anualmente planificación de capacitación para su personal, el mismo que se contempla en las partidas correspondientes dentro del presupuesto anual. Cada Unidad presenta sus respectivos requerimientos de capacitación y la Unidad de Recursos Humanos es quien ejecuta los planes de capacitación y mantiene una bitácora de los mismos.

Para el desarrollo de aplicaciones para el área de Artes Gráficas, Gestión Tecnológica planifica cursos puntuales que cubren los requerimientos de desarrollo de proyectos técnicos.

El personal que forma parte de Gestión Tecnológica para el área de Artes Gráficas debe cumplir con ciertos requisitos y si no los tienes debe pasar previamente con un proceso de capacitación.

3.2.1.4. KPA 22: INTEGRACIÓN DEL AMBIENTE ORGANIZACIONAL

Bajo el contexto de que el ambiente organizacional tiene como fin que el Producto, el Proceso de infraestructura de desarrollo, y la administración del personal sea integrado el IGM aplica este concepto en plan estratégico ,pues a través de este la institución guía su funcionamiento en el mediano y largo plazo y permite proyectar y ordenar sus actividades, así como apoyar los cambios y modificaciones que impulsen su desarrollo y permitir involucrarse en el proceso de consecución de una Institución que de respuestas, oportunas, eficientes y eficaces a las exigencias que le plantea el entorno.

Este plan estratégico se basa en lo siguiente:

3.2.1.4.1. METODOLOGÍA

La metodología desarrollada para la elaboración de la propuesta de desarrollo institucional está basada en una estructura lógica de objetivos prioritarios claramente identificados e interrelacionados, dotados de indicadores óptimos de medición que faciliten su planificación, ejecución, control y toma oportuna de decisiones.

3.2.1.4.2. MISIÓN

Somos el organismo autorizado por el Estado Ecuatoriano para generar y regular la información y bases de datos Cartográfica, Geográfica del país, proveer soluciones gráficas y de seguridad documentaria; extensión cultural en el campo científico de la astronomía y ciencias afines, que fortalecido con personal calificado, tecnología de vanguardia, procesos de mejoramiento continuo y respeto al medio ambiente. Contribuye con el desarrollo nacional.

3.2.1.4.3. VISIÓN

Satisfacer a los clientes a nivel nacional con proyección internacional, mediante soluciones integrales de cartografía, geografía, artes gráficas y seguridad documentaria, basados en una cultura de calidad y respaldados en la investigación técnica y científica.

3.2.1.4.4. OBJETIVOS EMPRESARIALES – ESTRATEGIAS

3.2.1.4.4.1. OBJETIVO ESTRATÉGICO Perspectiva Cliente:

Satisfacer y retener a los clientes/usuarios es vital para el crecimiento y fortalecimiento institucional, por lo que debe darse énfasis en las acciones

estratégicas de supervisión y monitoreo que permitan garantizar la satisfacción de los clientes mediante las siguientes metas estratégicas:

- Posicionar la imagen institucional
- Mejorar la satisfacción del cliente
- Generar una adecuada difusión cultural

3.2.1.4.4.2. OBJETIVO ESTRATÉGICO Perspectiva Interna:

Está relacionada al .3 desarrollo de acciones estratégicas por parte del IGM que permitan adquirir un peso específico institucional desarrollando capacidades técnicas y competencias como autoridad técnica, con el fin de mejorar su poder de negociación, dentro de una nueva estructura organizacional, considerando las necesidades de aprendizaje organizacional que se generan en una reestructurada institución.

Las metas estratégicas desde esta perspectiva son:

- Incrementar la productividad y el mejoramiento continuo.
- Unificación y aprobación de la normativa técnica.
- Aumentar la participación de mercado.
- Mantener información actualizada.
- Generar alianzas estratégicas.
- Mejorar la relación con proveedores.
- Implementar sistemas de seguridad integral.
- Provocar cambios y emisión de Leyes.
- Generar y mantener una cultura de calidad total/Mejorar la competitividad.
- Ser ambiental y socialmente responsables.

3.2.1.4.4.3. OBJETIVO ESTRATÉGICO Perspectiva Desarrollo Humano y Tecnología:

Generar desde los distintos ambientes del conocimiento, procesos de aprendizaje para el desarrollo del pensamiento científico y tecnológico, comprometiendo a cada una de las áreas del IGM, a asumir su propia función y responsabilidad (seguridad y ambiente), buscando la realización personal y profesional a través del desarrollo de principios como: autoestima, el respeto, el reconocimiento a la diferencia y la búsqueda de un mejor bienestar.

Las metas estratégicas desde esta perspectiva son:

- Mantener personal capacitado y motivado
- Mejorar el clima laboral
- Fortalecer la gestión del talento humano y la seguridad industrial
- Mejorar la comunicación interna
- Optimizar las tecnologías de información

3.2.2. ÁMBITO PRODUCTO DE SOFTWARE

El desarrollo de las KPAs de este Ámbito se encuentran directamente relacionados con el proceso de desarrollo del sistema informático requerido para el desenvolvimiento del proyecto. Además las actividades dentro de cada KPA de este Ámbito se constituyen repetitivas para cada SPRINT definido dentro del proyecto, en esta sección se procederá a documentar únicamente las actividades correspondientes al SPRINT No.1, las iteraciones del resto de SPRINTS definidos en el presente trabajo se los ubicará en la sección de Anexos.

3.2.2.1. KPA 1: GESTION DE REQUERIMIENTOS

El primer aspecto básico a cubrir es la Gestión de Requerimientos, se debe tener cuidado en el desarrollo de esta actividad puesto que aquí se sentará los objetivos y metas del proyecto.

3.2.2.1.1. GESTIÓN DE LA PILA DEL PRODUCTO

En la tabla 3.2 se muestra la definición del equipo de trabajo y roles de cada integrante en el caso de estudio.

ROL DEL PROYECTO	NOMBRE RESPONSABLE	CARGO
Dueño del Producto	Ing. Rómulo Garzón	Jefe de Producción de la Gestión Artes Gráficas del IGM
Scrum Master	Ing. Mercedes Játiva	Coordinador Gestión Tecnológica IGM.
Desarrollador 1	Ing. Pablo Sinchiguano	Desarrollo Gestión Tecnológica
Desarrollador 2	Ing. Rina Pacheco	Desarrollo Gestión Tecnológica
Usuarios	Ing. Marcelo Lobato	Coordinador Informática Centro de Movilización Pichincha.
Gerentes	CrnI. Iván Acosta A.	Director Instituto Geográfico Militar

Tabla 3.2: Definición de roles del proyecto
Elaborado por: M. Játiva, J. Carrera

La tabla 3.3 muestra en forma general los elementos de la Pila de Producto.

ID	Elemento	Importancia	Est. Inicial	Estado
001	Captura Datos Gráficos de Ciudadano	10	9	Inicial
002	Identificación del registro del ciudadano	10	5	Inicial
003	Formatos	9	12	Inicial
004	Autorización reimpressiones	6	7	Inicial
005	Conexión a base de datos	6	1	Inicial

006	Manejo de Paquetes	6	2	Inicial
007	Parametrización de Formatos	6	2	Inicial

Tabla 3.3: Pila de Producto Visión General
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.4 se detalla en elemento 001 de la Pila de Producto.

001	Captura Datos Gráficos
Importancia	10
Estimación Inicial	9
Como probarlo	<p>Verificar el correcto formato de las imágenes capturadas.</p> <ul style="list-style-type: none"> • Fotografía: archivo tipo JPG, dimensiones 400x400, máximo tamaño en disco 12 KB. • Huella dactilar: archivo tipo JPG, dimensiones 600x4080, máximo tamaño en disco 25 KB. • Firma digital: archivo tipo JPG, dimensiones 300x600, máximo tamaño en disco 8 KB. • Código de barras bidimensional: archivo tipo JPG, dimensiones 400x800, máximo tamaño en disco 30 KB, datos contenidos Cédula, Grado, Identificación Militar, fecha emisión.
Descripción	<p>Las imágenes capturadas correspondientes a fotografía del ciudadano, huella digital, firma y código de barras bidimensional deben ser almacenadas en la base de datos.</p> <p>Se debe realizar una interfaz que ofrezca la funcionalidad al operador de manejar todos los dispositivos periféricos de captura y generación de código de barras bidimensional.</p> <p>Las imágenes capturadas por medio de los dispositivos periféricos, deben ser sometidas a procedimientos gráficos implementados en la aplicación para obtener como resultado las imágenes con las características descritas previo su almacenamiento en la base de datos.</p> <p>Por cuestiones de almacenamiento es indispensable que las</p>

	imágenes cumplan las características de formato establecidas.
Estado	Inicial

Tabla 3.4: Elemento 001 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.5 se detalla en elemento 002 de la Pila de Producto.

002	Identificación del registro del ciudadano
Importancia	10
Estimación Inicial	5
Como probarlo	Realizar el proceso de registro completo de los datos del ciudadano con el sistema DIRMOV y verificar que el sistema IGM permita la captura de datos gráficos correspondiente. Verificar que el sistema IGM le asigne la tarjeta de identificación que le corresponde al ciudadano.
Descripción	El sistema para la emisión de Tarjetas de Identificación Militar debe enlazarse con el sistema de registro de la Dirección de Movilización. El sistema desarrollado por el IGM debe dar la posibilidad al operador de seleccionar el registro de ciudadano pendiente a la captura de datos gráficos para proceder a la emisión de la respectiva tarjeta de identificación. El sistema IGM debe ser capaz de diferenciar el tipo de tarjetas que le corresponde a un ciudadano. Se debe permitir la búsqueda del registro de ciudadano por: <ul style="list-style-type: none"> • Número de Cédula. • Número de comprobante. • Apellidos.
Estado	Inicial

Tabla 3.5: Elemento 002 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.6 se detalla en elemento 003 de la Pila de Producto.

003	Formatos
------------	-----------------

Importancia	9
Estimación Inicial	12
Como probarlo	Verificar que el formato gráfico en la emisión de la tarjeta de identificación corresponda al diseño impreso aprobado por la Dirección de Movilización.
Descripción	<p>Cada formato de Tarjeta de Identificación Militar debe ser implementado a nivel de sistema. Este formato es utilizado para ser impreso en forma personalizada para un ciudadano.</p> <p>Existen 4 tipos de tarjetas de Identificación Militar:</p> <ul style="list-style-type: none"> • Tarjeta de Militar Activo. • Tarjeta de Militar Pasivo. • Cedula Militar. • Tarjeta de Dependiente <p>Cada tarjeta posee un formato específico propio y este formato debe ser implementado en el sistema para la emisión física del documento.</p>
Estado	Inicial

Tabla 3.6: Elemento 003 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.7 se detalla en elemento 004 de la Pila de Producto.

004	Autorización de Reimpresiones
Importancia	6
Estimación Inicial	7
Como probarlo	<p>Verificar que un registro de ciudadano quede bloqueado una vez emitido la correspondiente tarjeta de Identificación Militar.</p> <p>Verificar el desbloqueo del registro una vez ingresada la autorización de reimpresión.</p> <p>Verificar el correcto registro de autorización de reimpresión en la base de datos.</p>
Descripción	Una vez emitida la Tarjeta de identificación Militar el

	<p>correspondiente registro deberá tomar el estado de bloqueado, de tal forma que no sea posible realizar más emisiones físicas para ese registro. Sin embargo existen ocasiones en las que es necesario por diferentes razones el realizar una nueva impresión de la Tarjeta de Identificación con un registro, esta reimpresión debe ser aprobada por el jefe del centro.</p> <p>El sistema debe poseer la funcionalidad de bloque de registro, desbloqueo de registro una vez autorizada la reimpresión y registro de autorización de reimpresión.</p>
Estado	Inicial

Tabla 3.7: Elemento 004 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.8 se detalla en elemento 005 de la Pila de Producto.

005	Conexión a base de datos
Importancia	6
Estimación Inicial	1
Como probarlo	Probar la conexión del sistema a una base de datos principal y a una base de datos de respaldo.
Descripción	<p>El sistema debe poder parametrizar la conexión a la base de datos, esta conexión debe ser seleccionada por el operador del sistema.</p> <p>Se debe implementar dos alternativas de conexión, base principal y base de respaldo.</p>
Estado	Inicial

Tabla 3.8: Elemento 005 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.9 se detalla en elemento 006 de la Pila de Producto.

006	Manejo de Paquetes
Importancia	6
Estimación Inicial	2

Como probarlo	<p>Registrar un paquete de tarjetas y proceder a la asignación a un usuario operador del sistema.</p> <p>Simular el proceso de emisión de tarjeta y registro de daño, comprobar el guardado de registro en la base de datos.</p>
Descripción	<p>La especie para impresión de la Tarjeta de Identificación Militar, se encuentra numerada y cada numeración corresponde a un paquete.</p> <p>El sistema debe registrar la asignación de un paquete de 100 tarjetas “vírgenes” para personalización a un operador, el operador se hace responsable por cada tarjeta recibida del paquete.</p> <p>El operador debe registrar las fallas de impresión y confirmar las tarjetas correctamente personalizadas, de esta forma al final del día se debe hacer un cierre de tarjetas emitidas con tarjetas con daño y tarjetas en blanco.</p>
Estado	Inicial

Tabla 3.9: Elemento 006 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

En la tabla 3.10 se detalla en elemento 007 de la Pila de Producto.

007	Parametrización de Formatos
Importancia	6
Estimación Inicial	2
Como probarlo	Cambiar valores de parametrización y verificar que el cambio sea reflejado en la impresión de cada formato.
Descripción	<p>Cada formato de Tarjeta de Identificación Militar contiene su propio formato, referente a:</p> <ul style="list-style-type: none"> • Color de la barra gráfica impresa bajo el título. • Escudos de armas de cada rama. • Posicionamiento de letras en la personalización. • Datos del director firmante.

	Las variables de formato anteriormente mencionadas, deben poder ser actualizadas por un usuario a nivel de aplicación.
Estado	Inicial

Tabla 3.10: Elemento 007 Pila de Producto
Elaborado por: M. Játiva, J. Carrera

Cada tabla de elemento será almacenadas en un documento de Excel con el nombre de BackLog.xls, estará compartido y a disposición del equipo de desarrollo en una unidad de red, la ubicación de este recurso es la siguiente: \\IGM07\Proyectos\Desrrollo\ArtesGraficas\Tecnicos\DIRMOV\BackLog.xls.

3.2.2.1.2. SEGUIMIENTO DE REQUERIMIENTOS DE LA PILA DEL PRODUCTO.

En esta sección según se establece en el Modelo de Gestión, se ha procedido a realizar la correspondiente revisión de cada elemento de la Pila del Producto definido en la sección anterior, determinando que:

- La conclusión de cada elemento es evaluable.
- Es posible asignar una estimación inicial para cada elemento de la pila.
- La implementación de cada elemento es relativamente independiente unos de otros.
- No existen elementos demasiados grandes, esta es una apreciación subjetiva, pero se ha determinado como parámetro en base al tiempo de desarrollo establecido para el presente proyecto, que la estimación inicial de cada elemento de la Pila del Producto no sea mayor a 10.

Con el establecimiento de estos parámetros, se da inicio a la actividad de seguimiento de los requerimientos de la Pila del Producto. Como parte del seguimiento, se debe considerar que cada cambio en un elemento será registrado en campo Estado correspondiente.

3.2.2.1.3. CONGELACIÓN DE REQUERIMIENTOS DURANTE CADA SPRINT.

Esta sección sienta una práctica inalterable que será tomada en cuenta durante cada Sprint a lo largo del proyecto.

Los requerimientos no pueden ser alterados durante un Sprint, y en caso de ser completamente necesario la modificación de uno u otro requerimiento, se lo hará conforme lo estipulado en el Modelo de Gestión y quedará documentado en esta sección y reflejado en la Pila del Producto, en el elemento respectivo.

En la implementación del caso de estudio no se ha encontrado necesidad de realizar ningún cambio en la definición de requerimientos.

3.2.2.1.4. INTEGRACION CONTINUA

Se asegurará la correcta funcionalidad de cada demo con la utilización de la herramienta de integración SUBVERSION, con esta herramienta se realizará el control de versiones y actividades de integración en el presente trabajo, Subversión posee las siguientes características:

- Sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones son atómicas.
- El repositorio es posible accederlo de forma local, se puede configurar seguridades para el control de acceso a los archivos.
- Es posible utilizar herramientas gráficas para el manejo del repositorio.

Se ha seleccionado la herramienta por cuestiones de facilidad de uso y consideraciones de compatibilidad con el lenguaje de desarrollo empleado en la implementación del proyecto objeto del caso de estudio.

3.2.2.1.5. INTERACCIÓN CONTINUA CON EL CLIENTE

En esta sección se establece que la reunión de Planificación de Sprint será llevada a cabo previo el inicio de cada Sprint con participación de los roles mostrados en la tabla 3.11.

Rol	Nombre	Obligatorio
Dueño Producto	Ing. Rómulo Garzón	SI
Scrum Master	Ing. Mercedes Játiva	Si
Desarrollador 1	Ing. Pablo Sinchiguano	SI
Desarrollador 3	Ing. Rina Pacheco	SI
Usuario	Ing. Marcelo Lobato	No

Tabla 3.11: Participantes Planificación de Sprint
Elaborado por: M. Játiva, J. Carrera

El Scrum diario será llevado a cabo en la sala de reuniones de la Gestión Tecnológica ubicado en el segundo piso del edificio principal del Instituto Geográfico Militar. La hora de inicio es las 9 de la mañana y no se podrá extender más de las 9H30, los asistentes a cada Sprint se muestran en la tabla 3.12.

Rol	Nombre	Obligatorio
Dueño Producto	Ing. Rómulo Garzón	NO
Scrum Master	Ing. Mercedes Játiva	Si
Desarrollador 1	Ing. Pablo Sinchiguano	SI
Desarrollador 3	Ing. Rina Pacheco	SI
Usuario	Ing. Marcelo Lobato	No

Tabla 3.12: Participantes Scrum diario
Elaborado por: M. Játiva, J. Carrera

Las reuniones definidas en esta sección serán ejecutadas a durante la duración del proyecto definiendo así la interacción continua con el cliente, que en este caso es ocupado por el Ing. Rómulo Garzón en su calidad de Jefe de Producción de la Gestión Artes Gráficas del Instituto Geográfico Militar.

3.2.2.2. KPA 2: PLANIFICACIÓN DE PROYECTOS DE SOFTWARE

3.2.2.2.1. PLANIFICACIÓN SCRUM-XP AL INICIO DE CADA PROYECTO.

Se realiza la Reunión de Planificación del Proyecto con fecha 3 de octubre del 2011, de ésta reunión se obtiene como resultado el Acta de Planificación del Proyecto “EMISION DE TARJETAS DE IDENTIFICACIONES MILITARES DE LAS FUERZAS ARMADAS DIRMOV”, y la Pila del Producto Inicial.

El acta es mostrada en la sección de anexos. Como resultado de la reunión de Planificación se obtiene la definición de la Pila del Producto, ésta pila se encuentra documentada en la sección anterior en el cubrimiento de la KPA 1.

3.2.2.2.2. PLANIFICACIÓN DE SPRINT AL INICIO DE CADA UNO.

En esta sección se ha asegurado que se cumpla lo que se muestra en la tabla 3.13.

Elemento	Cumple (SI/NO)
¿Existe la Pila de Producto y está terminada?	SI
¿Todos los elementos de la Pila de Producto poseen valores de importancia?	SI
¿Se encuentran documentados cada elemento de la Pila del Producto de tal forma que el Dueño del Producto entiende su significado?	SI

Tabla 3.13: Verificación para planificación de Sprint
Elaborado por: M. Játiva, J. Carrera

Para efectos del presente trabajo, en esta sección se documentará únicamente la Reunión de Planificación de un Sprint, en la sección de anexos se podrá encontrar la documentación de los Sprints restantes.

3.2.2.2.1. DEFINICIÓN DE LA META DEL SPRINT

La tabla 3.14 muestra la meta definida para el Sprint 1.

SPRINT No.	1
Nombre SPRINT:	“Captura de Datos gráficos de Ciudadano”
Meta de SPRINT:	<p>Capturar y almacenar la base de datos Sybase las imágenes del ciudadano a quien se le emitirá la tarjeta de identificación militar, estas imágenes corresponden a:</p> <ul style="list-style-type: none"> • Fotografía. • Huella digital. • Firma. <p>Generar código de barras bidimensional personalizado para cada usuario.</p> <p>Realizar la búsqueda del registro de ciudadano por:</p> <ul style="list-style-type: none"> • Número de cédula. • Número de comprobante. • Apellidos

Tabla 3.14: Meta de Sprint No 1
Elaborado por: M. Játiva, J. Carrera

3.2.2.2.2. DEFINICIÓN DE LA LISTA DE MIEMBROS DEL EQUIPO DE TRABAJO Y EL NIVEL DE DEDICACIÓN

La tabla 3.15 muestra la definición del equipo de trabajo para el Sprint 1.

SPRINT No:	1
Nombre de SPRINT	“Captura de Datos gráficos de Ciudadano”

Duración del Sprint	15Días
Días laborables en el Sprint	14días

Nombre	Días disponibles durante Sprint	Horas Disponibles por día	Total horas disponibles
Pablo Sinchiguano	14	7	98
Rina Pacheco	14	7	98

Tabla 3.15: Definición Equipo de Trabajo para Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

La tabla 3.16 se define la división de Historias para el Sprint No. 1.

3.2.2.2.3. INCLUSIÓN DE HISTORIAS EN EL SPRINT(Pila de SPRINT)

No. SPRINT: 1				
Nombre SPRINT: Captura de Datos Gráficos de Ciudadano				
Id.	Nombre Elemento	Tarea de Srpint	Responsables	Estimación I.
001	Captura Datos Gráficos de Ciudadano	Desarrollo librería de control Cámara Digital Cannon i8	Pacheco	1
		Desarrollo librería de control Pad de firmas	Pacheco	1
		Desarrollo librería de control lector de huellas dactilares	Pacheco	1
		Desarrollo de función para tratamiento de imágenes.	Pacheco	1
		Transformación de imágenes al representación hexadecimal	Pacheco	1
		Acoplamiento de librerías a c#.net para generación de códigos bidimensionales.	Pacheco	1
		Actualización de registro en base de datos campos de imágenes del ciudadano seleccionado y Bbloqueo de registro para nueva impresión..	Pacheco	1
002	Identificación del registro de ciudadano	Búsqueda y despliegue por número de cedula	Pacheco	1
		Búsqueda y despliegue por número de comprobante	Pacheco	1
		Búsqueda y despliegue por apellidos	Pacheco	1
		Identificación de tarjeta que le corresponde al ciudadano	Pacheco	1
		Total:	Total:	11

Tabla 3.16: División de Historias para Sprint No. 1

Elaborado por: M. Játiva, J. Carrera

3.2.2.2.4. DEFINICIÓN DE LA DURACION DEL SPRINT

La duración de este SPRINT está determinada a 15 días calendario, 11 días para el desarrollo de las actividades propias de codificación, 3 días destinados para actividades complementarias contempladas en el presente trabajo.

La valor de duración es determinado en función a las estimaciones iniciales asignadas a cada elemento de la Pila de Producto que han sido asignados a la Pila de Sprint.

3.2.2.2.5. DEFINICIÓN DEL SITIO Y LA HORA PARA EL SCRUM DIARIO

En el caso de los Proyectos Técnicos Informáticos de Artes Graficas, se definirá el Scrum diario a las 9 de la mañana y el sitio asignado será la sala de reuniones de la Gestión Tecnológica del IGM.

3.2.2.2.6. ELABORACIÓN DE HISTORIAS TÉCNICAS

Para el SPRINT No. 1 se han definido las historias técnicas mostradas en la tabla 3.17, tabla 3.18 y tabla 3.19.

No. SPRINT: 1	
Nombre SPRINT: Captura de Datos Gráficos de Ciudadano	
No. 1	Instalación Servidor Base de Datos
Descripción: Instalar una Motor de Base de Datos en un ambiente similar al que se poseerá en producción.	
Recursos:	
<ul style="list-style-type: none"> • Equipo Servidor Base de Datos. 	

<ul style="list-style-type: none"> • Instaladores Sybase versión 8.0 • Manuales de Instalación.
Tiempo Requerido: 4 horas

Tabla 3.17: Instalación de Base de Datos Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

No. SPRINT: 1	
Nombre SPRINT: Captura de Datos Gráficos de Ciudadano	
No. 2	Instalación y configuración Periféricos
<p>Descripción: Instalar y configurar periféricos en los equipos de desarrollo:</p> <ul style="list-style-type: none"> • Lector de huellas dactilares. • Cámara de fotos Cannon. • Panel de firmas. • Impresora Zebra para PVC. 	
<p>Recursos:</p> <ul style="list-style-type: none"> • Lector de huellas dactilares • Instaladores drivers del equipo lector de huellas dactilares • Cámara fotográfica Cannon. • Instaladores drivers del equipo lector de la cámara Cannon. • Panel de firmas. • Instaladores drivers del panel de firmas • Impresora Zebra para PVC. • Instaladores drivers para la impresora Zebra. • Manuales de Instalación. 	
Tiempo Requerido: 2 horas	

Tabla 3.18: Instalación de Periféricos Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

No. SPRINT: 1
Nombre SPRINT: Captura de Datos Gráficos de Ciudadano

No. 3	Instalación y configuración de los clientes Sybase
Descripción: Instalar y configurar clientes Sybase para Windows en los equipos de desarrollo para acceso a la base de datos..	
Recursos: <ul style="list-style-type: none">• Instaladores Cliente Sybase compatible con Windows.• Manuales de instalación y configuración.	
Tiempo Requerido: 1 hora	

Tabla 3.19: Instalación y Configuración clientes Sybase Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

3.2.2.3. KPA 3: SEGUIMIENTO Y SUPERVISIÓN DE PROYECTOS DE SOFTWARE.

En esta sección del trabajo se registrará y documentará en forma general el proceso de Scrum diario del Sprint número 1 “Captura de Datos gráficos de Usuario”, las respectivas Actas de Sprint están disponibles en la sección de anexos.

3.2.2.3.1. SCRUM DIARIO SPRINT No. 1.

La tabla 3.20 muestra el proceso de Scrum diario para el Sprint No.1.

SPRINT No.: 1			
Nombre de SPRINT: “Captura de Datos gráficos de Ciudadano”			
No.	Fecha	Realizado desde el último Sprint	A Realizar hasta el próximo Sprint
1	4/10/2011	Generación del ambiente de desarrollo: <ul style="list-style-type: none"> • Instalación y configuración Servidor de base de datos. • Instalación y configuración de periféricos en máquinas de desarrollo. • Instalación de clientes Sybase en máquinas de desarrollo. • Configurar conexión a base de datos desde máquinas de desarrollo. • Instalación Framework de desarrollo en máquinas clientes 	Instalar y compilar librerías de desarrollo para los equipos periféricos (Cámara, Pad de firmas, Lector de huellas, Impresora) en la herramienta de desarrollo.
			Observaciones Se requiere disponer de los equipos periféricos (Cámara, Pad de firmas, Lector de huellas, Impresora) para el desarrollo.

2	5/10/2011	<p>Instalación y compilación de librerías de desarrollo para los equipos periféricos (Cámara, Pad de firmas, Lector de huellas, Impresora).</p> <p>Verificación de compatibilidad de las librerías (dlls) con Visual Studio 2008</p>	<ul style="list-style-type: none"> • Encontrar librerías compatibles de desarrollo del equipo lector de huellas para VS2008. • Generar interfaz de captura de imágenes en línea con la Cámara Cannon. • Generar interfaz de captura de imágenes en línea con Pad de firmas. 	<p>Librerías de desarrollo para el equipo lector de huellas no es compatible con VS2008</p>
3	6/10/2011	<p>Generación de interfaz de captura de imágenes en línea con la Cámara Cannon.</p> <p>Tratamiento de Imágenes en cuanto a dimensiones físicas.</p> <p>Generación de librería para el corte de fotografías.</p>	<p>Conversión de imágenes a su representación en cadenas de caracteres hexadecimales para almacenamiento en la base de datos Sybase.</p> <p>Compilar librerías de desarrollo del lector de huellas dactilares a C++, y generación de dll compatible con C# VS.net.</p> <p>Generar interfaz de captura de imágenes en línea con el lector de huellas dactilares.</p>	<p>Aún no es posible solucionar la compatibilidad de la librería de desarrollo del lector de huellas dactilares.</p>
4	7/10/2011	<p>Conversión de imágenes a su representación en cadenas de caracteres hexadecimales.</p> <p>Compilación de librerías de desarrollo del lector de</p>	<p>Implementar tratamiento de imágenes para cada imagen capturada (Fotografía, huella dactilar, firma).</p>	

		huellas dactilares a C++. Generación de dll compatible con C# VS.net. Generación de interfaz de captura de imágenes en línea con el lector de huellas dactilares.		
5	10/10/2011	Unificación de interfaz de captura de imágenes e implementación de función de tratamiento de cada una de estas imágenes.	Implementar función de guardado de imágenes en la base de datos Sybase. Generación de código de barras bidimensional.	
7	11/10/2011	Implementación e inclusión en interfaz la funcionalidad de generación de código de barras bidimensional personalizado.	Almacenar imágenes como un campo de una tabla de base de datos Sybase.	
8	12/10/2011	Implementación de Almacenamiento de imágenes en la base de datos según formato requerido.	Implementar búsquedas por los criterios establecidos.	
9	13/10/2011	Búsqueda y selección de registro de ciudadano por los parámetros de cédula, número de comprobante, apellidos.	Realizar bloqueo de registro emitido.	
10	14/10/2011	Bloqueo de registro después de la emisión de la correspondiente tarjeta.	Implementar y a asegurarse el direccionamiento al formato de tarjeta correcto.	
11	15/10/2011	Identificación de tarjeta que le corresponde al ciudadano	Realizar pruebas sobre el producto de SCRUM.	
12	15/10/2011	Pruebas de funcionalidad, actividades de		

		aseguramiento de la calidad.			
13	16/10/2011	Revisión Funcionalidad		Revisión Presentación y codificación	
14	17/10/2011	Revisión de Presentación y codificación.			

Tabla 3.20: Scrum diario Sprint No. 1

Elaborado por: M. Játiva, J. Carrera

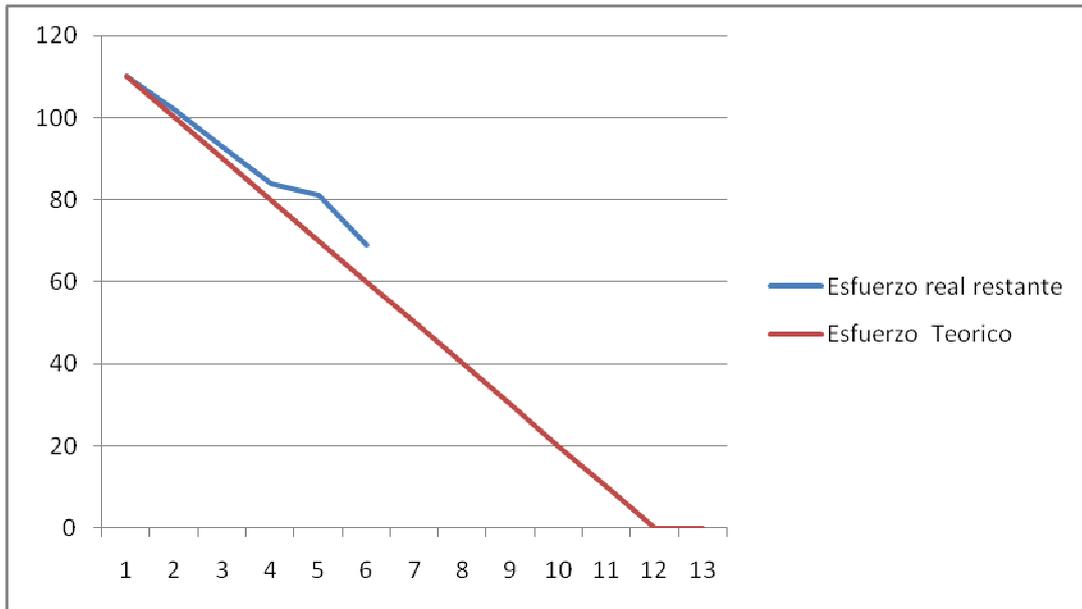


Figura 3.2: Avances Backlog Sexto día Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

En el primer día del Sprint, el equipo estimó que había aproximadamente 110 puntos de historia en los que se debe trabajar. Esta es, consecuentemente, la velocidad estimada para todo el Sprint. En el sexto día el equipo estima que quedan aproximadamente 65 puntos de historia por hacer. La línea de puntos muestra que el trabajo se encuentra levemente retrasado con respecto a la planificación.

La figura 3.3 muestra la grafica de Backlog resultante al final del Sprint No.1.

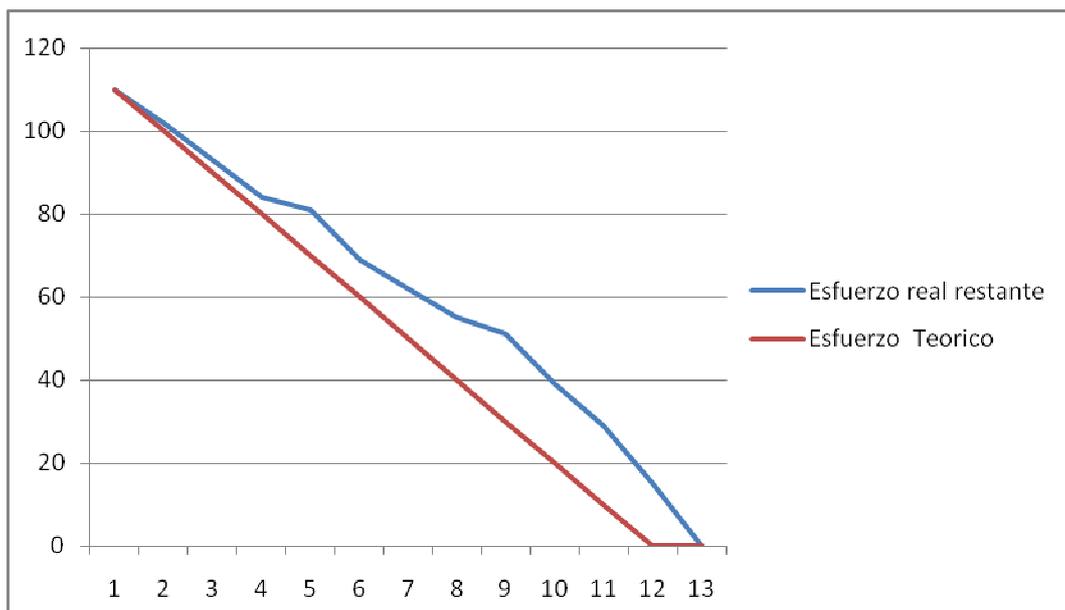


Figura 3.3: Avances Backlog al Final del Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

Si la línea de trabajo restante no está bajando hasta la finalización del trabajo cuando se acerca el final del Sprint, entonces el equipo necesita hacer ajustes, por ejemplo reducir el alcance del trabajo o encontrar una forma de trabajar más eficientemente al tiempo que mantienen un ritmo sostenible. Existe un pequeño desfase entre la planificación y la aplicación real según el gráfico lo cual se le considera como aceptable para el presente trabajo.

3.2.2.4. KPA 6: ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

3.2.2.4.1. PRUEBAS DE FUNCIONALIDAD

El desarrollo en pares asegura que el cumplimiento de estándares de codificación sean cubiertos en el proceso mismo, lo cual disminuye el tiempo destinado a las pruebas en este sentido, enfocándose directamente a las pruebas de funcionalidad.

La tabla 3.22 muestra la lista de verificación aplicada para la Revisión de Funcionalidad del producto obtenido al finalizar el Sprint No.1.

Revisión de Funcionalidad				
No. Sprint: 1				
Nombre SPRINT: “Captura de Datos gráficos de Usuario”				
Fecha: 15/10/2001				
Actividad	Si	No	No aplica	Información adicional
¿El producto cubre con la funcionalidad especificada en la Pila de Sprint?	X			
¿Los datos de la forma cambian en forma sincronizada?	X			
¿El tiempo de respuesta es razonable para el usuario?	X			
Cuando se cambia el valor de un campo de entrada, ¿se modifica también el campo de despliegue?	X			

Los bloques hijos están coordinados con el bloque padre en consulta, borrado y cuando se limpia la forma?	X			
Los campos que hacen referencia a datos de tablas ¿tienen cada uno su lista de valores?	X			
¿Las listas de valores son lentas para recuperar la información?		X		
¿El orden de navegación de los campos es el correcto?	X			
¿Se ha implementado manejo de excepciones adecuadamente?	X			
¿Los campos Validate from LOV funcionan adecuadamente?	X			
¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?			X	
¿Si llama reportes, la extensión de los reportes es la correcta? (NO rdf, debe estar sin extensión).			X	
¿El desarrollo de la funcionalidad objeto de la revisión de este Sprint se encuentra debidamente integrado?	X			

Tabla 3.22: Revisión de la Funcionalidad Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

3.2.2.4.2. REVISIÓN DE PRESENTACIÓN

La tabla 3.23 muestra la lista de verificación aplicada para la Revisión de Presentación del producto obtenido al finalizar el Sprint No.1.

Revisión de Presentación				
No. Sprint: 1				
Nombre SPRINT: “Captura de Datos gráficos de Usuario”				
Fecha: 15/10/2001				
Actividad	Si	No	No aplica	Información adicional
¿Están claramente definidos los bloques de información (Frames)?	X			
¿Tiene los encabezados de título y nombre de aplicación correctos?	X			
¿Las etiquetas de los campos son claras y representativas?	X			

¿Los campos de despliegue están completamente inhabilitados y del color respectivo?	X			
¿Los campos de solamente despliegue están claramente identificados?	X			
¿Los campos fecha tienen el formato DD-MON-RRRR y se puede ingresar los datos como Ej: 12ago2001?			X	Se toma la fecha del sistema para cualquier transacción.
Cuando se tiene una forma con múltiples tabs, ¿se conoce cuál es el registro padre de los tabs?	X			
¿Los Radio Groups tienen un frame que los abarca?	X			
¿Los campos están alineados en forma correcta?	X			
¿Los campos requieren y tienen Tooltip?			X	No se encuentra dentro de los estándares como obligatorio el implementar tooltip para un campo.

Tabla 3.23: Revisión de Presentación Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

3.2.2.4.3. REVISIÓN DE ESTÁNDARES DE CODIFICACIÓN

La tabla 3.24 muestra la lista de verificación aplicada para la Revisión de Estándares de Codificación del producto obtenido al finalizar el Sprint No.1.

Revisión de Estándares de Codificación				
No. Sprint: 1				
Nombre SPRINT: "Captura de Datos gráficos de Usuario"				
Fecha: 15/10/2001				
Actividad	Si	No	No aplica	Información adicional
¿Se ha hecho revisión por pares?	X			
¿Se ha realizado el proceso de afinamiento sql?	X			
¿Se encuentra el código debidamente documentado?	X			Código comentado.
¿Tienen las variables nombres fáciles de entender?	X			

Tabla 3.24: Revisión de Estándares de Codificación Sprint No. 1
Elaborado por: M. Játiva, J. Carrera

3.2.2.5. KPA 7: GESTION DE LA CONFIGURACIÓN DEL SOFTWARE

La siguiente tabla muestra las actividades de identificación de los elementos a gestionar, la aceptación de la inclusión de cada elemento fue realizada en las distintas reuniones de Sprint (SCRUM diario), el equipo decide si es conveniente que un elemento deba ser incluido en la gestión de configuración.

A continuación en la tablas: 3.25, 3.26, 3.27, se muestran los elementos identificados sobre los cuales se debe hacer gestión de configuración en el Sprint 1:

No. Sprint: 1	
Nombre SPRINT: “Captura de Datos gráficos de Usuario”	
Nombre del Elemento:	Librería para Manejo de Cámara Canon
Identificador:	001
Autor:	Ing. Rina Pacheco
Tipo de Archivo:	Dll, librería C#.NET
Responsable del elemento:	Ing. Rina Pacheco
Almacenamiento:	\\IGM07\DIRMOV\VERSION\DLLCANON\
Descripción:	Librería que permite el control de la cámara Canon. Esta librería permite el control en línea de:
Gestión de Cambios	<p>Versión 1.0 (3/10/2011): Toma de fotografía con manejo de Zoom, genera la fotografía en formato JPG en el directorio que le sea especificado.</p> <p>Versión 1.5(10/102011): Se agrega la funcionalidad para manejo de luz en la librería.</p> <p>Versión 2.0(11/20/2011):Se agrega la funcionalidad de manejo del formato de imagen generada, para el proyecto DIRMOV se establecer a JPG.</p>
Acceso:	Acceso de control total al usuario rpacheco

Tabla 3.25: Elemento de Gestión Librería para Manejo de Cámara Canon
Elaborado por: M. Játiva, J. Carrera

No. Sprint: 1	
Nombre SPRINT: "Captura de Datos gráficos de Usuario"	
Nombre del Elemento:	Librería para Manejo de Pad de firmas
Identificador:	002
Autor:	Ing. Rina Pacheco
Tipo de Archivo:	Dll, librería C#.NET
Responsable del elemento:	Ing. Rina Pacheco
Almacenamiento:	\\IGM07\DIRMOV\VERSION\DLLPADFIRMAS\
Descripción:	Librería que permite el manejo en línea del Pad de firmas.
Gestión de Cambios:	<p>Versión 1.0 (3/10/2011): Toma la imagen de la firma. Almacena la imagen en formato JPG en un directorio especificado.</p> <p>Versión 1.5 (10/10/2011): Se adhiere la funcionalidad de manejo del grosor de línea capturado en el pad de firmas y la opción de borrar el dibujo para iniciar nuevamente la captura.</p>
Acceso:	Acceso de control total al usuario rpacheco

Tabla 3.26: Elemento de Gestión Librería para Manejo de Pad de firmas TOPAZ
Elaborado por: M. Játiva, J. Carrera

No. Sprint: 1	
Nombre SPRINT: "Captura de Datos gráficos de Usuario"	
Nombre del Elemento:	Librería para Manejo de Lector de Huellas Dactilares
Identificador:	003
Autor:	Ing. Rina Pacheco
Tipo de Archivo:	Dll, librería C#.NET
Responsable del elemento:	Ing. Rina Pacheco
Almacenamiento:	\\IGM07\DIRMOV\VERSION\DLLHUELLAS\
Gestión de Cambios:	Versión 1.0(4/10/2011): Toma la imagen de la huella dactilar, permite el manejo del contraste de la imagen. Almacena la imagen en formato JPG en un directorio especificado.
Acceso:	Acceso de control total al usuario rpacheco

Tabla 3.27: Elemento de Gestión Librería para Manejo de Lector Huellas
Elaborado por: M. Játiva, J. Carrera

3.2.2.6. KPA8: DESARROLLO DE REQUERIMIENTOS

El desarrollo del proyecto ha sido dividido en 4 SPRINTs, al finalizar el tercer SPRITN se ha planificado una reunión de revisión de requerimientos para cumplir con esta KPA, esta reunión se realiza el 18 de noviembre con la participación de todo el equipo de desarrollo, el Scrum Master el Dueño de Producto. Se ha revisado en forma conjunta con los asistentes a mencionada reunión la definición de la Pila de Producto y se ha establecido como resultado que no es necesario realizar modificaciones ni establecer nuevos requerimientos para ser agregados a la Pila del Producto.

3.2.2.7. KPA10 INTEGRACION DEL PRODUCTO

Debido a que esta KPA se aplica a lo largo de fases de Scrum se realizará un control de la entrega de cada uno de los elementos de la pila de Sprints.

La tabla 3.28 muestra la lista de verificación utilizada para efectos de validación de la integración de Sprints.

Sprint de Entrega (Pila de Sprints)

No. SPRINT	Nombre	Si cumple	No cumple
Sprint 1	Captura de Datos gráficos de Ciudadano	X	
Sprint 2	Formatos	X	
Sprint 3	Parametrización	X	

Tabla 3.28: Integración de Sprints
Elaborado por: M. Játiva, J. Carrera

Si todos y cada uno de los SPRINTs tienen entregas que cumpla con los requerimientos del usuario, al realizar la integración completa se puede concluir que el producto final integrado cumple satisfactoriamente con los requerimientos del usuario.

3.2.2.8. KPA 11: VERIFICACIÓN.

La estrategia de Verificación de productos se lo realiza con la metodología de Verificación en pares, para lo cual se crea un ambiente de pruebas que utiliza la base de datos real pero graba en las base de pruebas, los pares serán distribuidos de la misma forma como se desarrollaron las interfaces, se deberá realizar las respectivas verificaciones a través de las diferentes interacciones de los sprints, cada vez que se presente un sprint terminado se aplicará las respectivas verificaciones hasta que no exista observaciones y se cubra todos los requerimientos del usuario.

La tabla 3.29 muestra le proceso de verificación número 2 aplicado al Sprint No.1.

VERIFICACIÓN DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Verificación No.:2					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Foto	Realizar búsqueda por cedula y capturar foto de la base de datos.	Si	Cuando se carga más de una vez la foto esta ya no se despliega en la interface. Aumentar el tamaño y calidad de la foto capturada. La foto es opaca implementar flash.
Verificado por: Ing Pacheco e Ing Sinchiguano					

Tabla 3.29: Verificación Sprint 1 verificación 2

Elaborado por: M. Játiva, J. Carrera

La tabla 3.30 muestra le proceso de verificación número 3 aplicado al Sprint No.1.

VERIFICACIÓN DE PRODUCTOS DE SPRINT					
--	--	--	--	--	--

Nombre de Sprint: 1		Fecha: 20-11-2011			
Verificación No.:3					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Huella digital	Realizar búsqueda por cedula y capturar huella y firma de la base de datos.	si	Existen muchos pasos para realizar la grabación de la huella , la captura debe ser más ágil
Verificado por: Ing Pacheco e Ing Sinchiguano					

Tabla 3.30: Verificación Sprint 1 verificación 3
Elaborado por: M. Játiva, J. Carrera

La tabla 3.31 muestra le proceso de verificación número 4 aplicado al Sprint No.1.

VERIFICACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1		Fecha: 20-11-2011			
Verificación No.:4					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Firma	Realizar búsqueda por cedula y capturar firma de la bdd	Si	Existen muchos pasos para realizar la grabación de la firma , la captura debe ser más ágil
					Al realizar búsquedas no se captura la firma
					Aumentar el visualizador de la firma pues no puede ver la imagen claramente.
					Mejorar la calidad de la imagen.

Verificado por: Ing Pacheco e Ing Sinchiguano

Tabla 3.31: Verificación Sprint 1 verificación 4
Elaborado por: M. Játiva, J. Carrera

La tabla 3.32 muestra le proceso de verificación número 5 aplicado al Sprint No.1.

VERIFICACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Verificación No.:5					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Foto	Realizar búsqueda por cedula y capturar foto de la bdd	Si	

Verificado por: Ing Pacheco e Ing Sinchiguano

Tabla 3.32: Verificación Sprint 1 verificación 5
Elaborado por: M. Játiva, J. Carrera

La tabla 3.33 muestra le proceso de verificación número 6 aplicado al Sprint No.1.

VERIFICACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Verificación No.:6 Iteracción No:2					
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Huella digital	Realizar búsqueda por cedula y caputurar huella y firma de la bdd	si	

Verificado por: Ing Pacheco e Ing Sinchiguano

Tabla 3.33: Verificación Sprint 1 verificación 6
Elaborado por: M. Játiva, J. Carrera

La tabla 3.34 muestra le proceso de verificación final aplicado al Sprint No.1.

VERIFICACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Verificación No.:2			Iteración No:2		
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
Verificado por: Ing Pacheco e Ing Sinchiguano					

Tabla 3.34: Verificación Sprint 1 Iteración 2
Elaborado por: M. Játiva, J. Carrera

3.2.2.9. KPA 12: VALIDACIÓN

Para las validaciones se trabaja con el Usuario final, en este proyecto se trabajará con el Coordinador Informática Centro de Movilización Pichincha, se realizarán pruebas paralelas en las instalaciones de la Dirmov con datos netamente reales, se utilizará cuadros de validaciones por cada iteración de Sprints.

La tabla 3.35 muestra la validación No. 1 aplicada al Sprint No.1.

VALIDACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 01-12-2011		
Validación No.:1			Iteración No:1		
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Foto	Realizar búsqueda por cédula y capturar foto de la bdd	Si	Falta Validar información existente en la base real para las búsquedas.

					Añadir Mensajes de Error
Validado por: Ing Marcelo Lobato					

Tabla 3.35: Validación 1 Sprint 1 Iteración 1
Elaborado por: M. Játiva, J. Carrera

La tabla 3.36 muestra la validación No. 2 aplicada al Sprint No.1.

VALIDACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Validación No.:2			Iteración No:1		
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Huella digital	Realizar búsqueda por cedula y capturar huella y firma de la bdd	si	Falta Validar información existente en la base real para las búsquedas.
Validado por: Ing Marcelo Lobato.					

Tabla 3.36: Validación 2 Sprint 1 Iteración 1
Elaborado por: M. Játiva, J. Carrera

La tabla 3.37 muestra la validación No. 3 aplicada al Sprint No.1.

VALIDACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 20-11-2011		
Validación No.:3			Iteración No:2		
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones

	Captura de Datos Gráficos del Ciudadano	Firma	Realizar búsqueda por cedula y capturar firma de la base de datos.	Si	Falta Validar información existente en la base real para las búsquedas.
Validado por: Ing Marcelo Lobato.					

Tabla 3.37: Validación 3 Sprint 1 Iteración 2
Elaborado por: M. Játiva, J. Carrera

La tabla 3.38 muestra la validación No. 2 aplicada al Sprint No.1.

VALIDACION DE PRODUCTOS DE SPRINT					
Nombre de Sprint: 1			Fecha: 01-12-2011		
Validación No.:4			Iteración No:3		
No.	Requerimiento Funcional	Entradas	Salida esperada	Cumple (SI/NO)	Observaciones
	Captura de Datos Gráficos del Ciudadano	Foto	Realizar búsqueda por cedula y capturar foto de la base de datos.	Si	
Validado por: Ing. Marcelo Lobato.					

Tabla 3.38: Validación 4 Sprint 1 Iteración 3
Elaborado por: M. Játiva, J. Carrera

3.2.2.10. KPA 16 GESTION DE RIESGOS

En la Unidad de Artes Gráficas del IGM se asignará directamente la gestión de riesgos del proyecto al equipo Scrum debido a que todos los proyectos técnicos

son asignados directamente al área de Tecnología y la responsabilidad del éxito recae sobre esta área.

La tabla 3.39 muestra en qué fase de Scrum se realiza la gestión de cada riesgo mencionado.

Riesgo	Cómo lo controla Scrum
Ampliación descontrolada de características	Scrum Inicial
Captura de requisitos mal realizada	Backlog
Calidad insuficiente	Sprint Review
Plazos optimistas impuestos	Plazos y Tiempos
Diseño inadecuado	Sprint Review y Sprint Retrospective.
Desarrollo orientado a la investigación	Avances en el ProductBurndown Chart y durante los Sprint Review.
Personal inadecuado	Scrum Diario
Fricción con los clientes	Intervención del ProductOwner en los Sprint Review.

Tabla 3.39: Gestión de Riesgos
Elaborado por: M. Játiva, J. Carrera

3.2.2.11. KPA 19: GESTION INTEGRADA DEL PROYECTO

La forma de trabajo definido para los Proyectos Informáticos Técnicos de la Gestión Artes Gráficas del IGM se define de la siguiente forma:

Se debe asignar un servidor virtualizado donde se implementará un Servidor de Integración Continua, residente en el centro de servidores ubicado en las oficinas de la Gestión Tecnológica del IGM, este computador está destinado exclusivamente para las tareas de integración. Cada par de desarrolladores trabaja en sus respectivas máquinas de desarrollo escribiendo su código y realizando sus casos de prueba individuales, comprobando y resolviendo cualquier error que se encuentre.

El rol de Gestor de Configuraciones es desempeñado por el jefe del equipo, este debe poner bajo control de versionado el código base inicial y cada par de desarrolladores debe utilizar el software de versionado, para de esta forma conservar las versiones del código base, estas actividades.

Además se debe instaurar el uso de comentarios a nivel de código y el uso de las historias de usuario (división de Sprint en tareas).

No es posible definir una herramienta de integración única puesto que los proyectos Informáticos Técnicos de la Gestión Artes Gráficas del IGM no son homogéneos y necesariamente para la selección de la herramienta que brindará el entorno de integración deberá ser realizada en base a las características propias de cada proyecto, tomando en cuenta principalmente características de compatibilidad.

En el presente trabajo se tomará Subversión como Herramienta de Integración principal, pudiendo variar como se explicó anteriormente en función de las necesidades que las características de cada proyecto determinen.

3.2.2.12. KPA 20: INTEGRACIÓN DEL EQUIPO O COORDINACIÓN INTERGRUPAL

La tabla 3.40 la coordinación intergrupala definida en pares de desarrollo para cada Sprint.

Scrum Master: Ing. Mercedes Játiva

Sprint	Nombre Sprint	Pares
1	Captura de Datos gráficos de Ciudadano	Ing. Pacheco Rina. Ing.Sinchiguano Pablo
2	Formatos.	Ing. Pacheco Rina. Ing. Sinchiguano Pablo
3	Parametrización	Ing. Pacheco Rina. Ing.Sinchiguano Pablo

Tabla 3.40: Coordinación Intergrupala
Elaborado por: M. Játiva, J. Carrera

El principal propósito de trabajar en pares es que el código desarrollado en el proyecto pertenezca al equipo y no a una sola persona, los integrantes que conforman el equipo tienen diferentes destrezas, al trabajar de esta forma se puede consolidar ciertas destrezas y en caso de no tenerlas plantear el camino para adquirirlas, es importante que todos y cada uno conozca a detalle el funcionamiento de cada uno de los módulos del proyecto para que el mismo pueda ser soportado por cualquiera de los miembros del equipo.

Adaptando los conceptos de Scrum en cuanto a la forma de organizar al equipo, se concibe al equipo como un todo en el cual no existe un jefe del Proyecto sino todos y cada uno asume la responsabilidad de sacar el proyecto en marcha, se maneja el concepto del Scrum Master quien estará encargado de verificar se vayan cumpliendo los requerimientos de la pila del producto y distribuirá los Sprints a los desarrolladores, así como también es quien negociará con el Dueño del Producto en caso de nuevos requerimientos o tiempos de entrega.

Adaptando los conceptos de XP en cuanto a desarrollo, éste Proyecto será realizado por dos pares de desarrolladores, se utilizará un servidor de aplicaciones que debe ser accesado por cada una de las parejas para realizar sus desarrollos bajo el manejo de versionamiento, el horario de desarrollo consistirá en trabajar mínimo 4 horas de corrido en un mismo tema en el cual un programador será quien digita y el otro deberá seguir la lógica del proceso. Una vez cumplida la meta se procederá con las pruebas de verificación las cuales serán aplicadas por la misma pareja.

En este Proyecto se ha planificado no realizar cambios de pares pues al ser un proyecto piloto en utilización de este concepto deberá aplicarse paulatinamente en cada fase y luego analizar si es exitosa o no.

CAPÍTULO 4 CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- El uso de una metodología es completamente necesario para el desarrollo con el cumplimiento de un orden en cualquier tipo de proyecto, sin embargo es necesario definir claramente y tomar en cuenta las características propias de la empresa y de los proyectos a desarrollar como requisitos para definir la metodología más adecuada a utilizar. En el caso de empresas en las que el desarrollo de software no constituye el enfoque principal de su negocio como las caracterizadas en este trabajo se debe considerar metodologías de desarrollo no tradicionales que se centran principalmente en cumplimiento del objetivo funcional del proyecto informático utilizando recursos reducidos, estas características ofrecen las metodologías de desarrollo Ágil.
- Las metodologías tradicionales de desarrollo son utilizadas en todo tipo de Proyectos y su formalidad exige un esfuerzo considerable en documentación y gerenciamiento de proyecto lo cual genera una sobrecarga que no encaja en proyectos pequeños y medianos, sin embargo aplicar una metodología ágil requiere de un cambio de paradigma en la manera de ver y hacer las cosas en todos los miembros del proyecto, tanto a nivel de gerencia, a nivel de desarrolladores y del cliente, lo cual es un reto muy ambicioso para una empresa pero no inalcanzable.
- No existe una metodología de desarrollo única que permita garantizar el éxito de cualquier proyecto de desarrollo de software. Toda metodología requiere de cierta adaptación al tipo de proyecto, al tamaño del mismo, al cliente y a la idiosincrasia propia de cada empresa. La aplicabilidad de una metodología debería ser analizada en cada caso evaluando sus respectivas ventajas y desventajas.

- El Modelo de Gestión de Calidad propuesto basado en utilizar metodologías ágiles Scrum y XP con el modelo de procesos CMMI puede ser considerado válido pues todas las áreas de KPAs de nivel de madurez 2 y un número significativo de KPAs de nivel de madurez 3 son soportadas a través de prácticas ágiles. Dentro del contexto del tipo de proyecto utilizado los métodos ágiles proporcionan buenas prácticas y combinando con el enfoque de CMMI pueden garantizar Calidad en el producto terminado.
- La relación de correspondencia definida entre actividades de Metodologías ágiles y KPAs de CMMI, establece una guía de apoyo para toda aquella organización que quiera utilizar metodologías ágiles y que a futuro desee certificarse de acuerdo al modelo de procesos CMMI. También puede utilizarse en forma inversa como guía de apoyo en el caso de que una organización que ya tiene implantado el modelo de procesos CMMI y necesite incorporar métodos ágiles en alguno de sus proyectos.
- La implantación de un modelo de calidad como CMMI en empresas con características similares al IGM detalladas en este documento se torna complejo puesto que estos modelos se encuentran direccionados a organizaciones complejas integradas por un gran número de personas, empero es posible definir y utilizar ciertas prácticas que se ajustan a la realidad de funcionamiento de una mediana empresa y aplicarlos en un proceso específico obteniendo resultados favorables en la calidad del producto final y durante el proceso de elaboración.
- La combinación de prácticas de metodologías ágiles para desarrollo de software con el Modelo de Capacidad de Madurez (CMMI), permite generar lineamientos y actividades para gestión de la calidad en procesos dinámicos con requerimientos cambiantes y tiempos cortos de desarrollo.
- En el modelo propuesto se definen aspectos organizacionales en las KPAs 13,14,15 y 22 mientras que las KPAs 1, 2,3,6,7,8,10,11,12,16,19 y 20

establecen actividades y procedimientos referentes a la parte operativa técnica que deben conducir a alcanzar las metas definidas en la parte organizacional, todos estos aspectos tanto organizacionales como operativos deben ser conocidos por todos los miembros de equipo de trabajo de una empresa para la correcta aplicación de modelo propuesto en este trabajo.

- El éxito de la aplicación del Caso de estudio utilizando el Modelo de Gestión de Calidad propuesta se atribuye a los siguientes factores: El personal de desarrollo de Gestión de Tecnología para Artes Gráficas mantiene su propia autonomía, lo que permitió eliminar la figura de jerarquías dentro del proyecto y consolidar el proyecto a todo el equipo de trabajo. El cliente fue un cliente externo de la institución el mismo que tenía un alto compromiso de cumplimiento de contrato y el tipo de proyecto fue un proyecto corto de tres meses de duración cuyos requerimientos estuvieron claramente definidos.
- La experiencia y características en cuanto a conocimientos técnicos del equipo de desarrollo en las herramientas de desarrollo y dispositivos especiales utilizados en el tipo de proyectos para el cual está referido el Modelo de Gestión propuesta en el presente trabajo, constituye un factor determinante en la consecución exitosa del proyecto.
- La presencia de jefes, gerentes o personas que ocupen cualquier puesto de autoridad en reuniones las del Scrum Diario provoca una sensación en el equipo de trabajo negativa, puesto que se crea la sensación de que cada miembro del equipo debe informar de un gran progreso todos los días, esto constituye una expectativa muy poco realista, pero provoca y cohibición en el informar sobre otro tipo de problemas. Es recomendable y más útil que los interesados en el proyecto se acerquen al equipo después de la reunión y se ofrezcan a ayudar con cualquier bloqueo que está haciendo disminuir el progreso del equipo.

- El trabajo en codificación en parejas planteado en el presente trabajo siguiendo la metodología XP, maximiza la calidad en la generación de código, lo que hace que el tiempo y recursos invertidos en pruebas y correcciones se minimice, esto es una de las principales ventajas del trabajo de codificación en parejas.
- Se evita ciclos de pruebas largos y contribuir a aumentar la calidad en el producto de software final, especificando Sprint pequeños, es decir no incluyendo demasiados elementos de la Pila del Producto en cada Sprint. Se ha observado que esta práctica disminuye la cantidad de errores y aumenta la productividad del equipo encargado del Sprint, a la vez que se disminuye los tiempos en los ciclos de pruebas.
- La utilización de herramientas de TI para manejo de Scrum y Xp permitieron visualizar de mejor forma las ventajas del uso de metodologías ágiles así como también incrementar valor agregado a la forma de realizar los procesos en la empresa convirtiendo tareas complejas en escenarios más simples.
- Es posible observar que la mejora en las prácticas de la administración de proyectos de desarrollo de software, dadas las características de la competitividad actual, solo puede lograrse por medio de la integración de múltiples herramientas, que van desde la creación de infraestructura y uso de Tecnologías de Información y Comunicaciones, hasta elaborar planes e iniciativas de mejora de calidad del proceso de desarrollo y fomentar el aprendizaje organizacional, uso de técnicas, y adopción de nuevos modelos de gestión que apoyen la estrategia de cada organización.
- En equipos pequeños de desarrollo, las habilidades experiencia y conocimiento de los miembros de los equipos juegan un papel fundamental en la obtención de productos de calidad dentro del tiempo establecido. El grupo no puede dedicar mucho tiempo a trabajo

administrativo, el tiempo debe ser enfocado en gran parte a trabajo de diseño, programación y pruebas de producto en construcción.

- Ningún cambio es simple, especialmente cuando esto involucra modificar comportamientos que se los tiene arraigados, acogerse al cambio es difícil inclusive cuando con las prácticas o comportamientos actuales existe descontento.

4.2. RECOMENDACIONES

- Se recomienda analizar la posibilidad de extender el modelo propuesto en este trabajo utilizando prácticas de metodologías de desarrollo no necesariamente ágiles en caso de aplicarlo en empresas o proyectos en los que las características que se describen en este documento difieran.
- Dado que las practicas ágiles no son comunes en el mercado, se sugiere estimular y fortalecer los conceptos en la teoría y la práctica a los estudiantes de las universidades, lo que permitiría entregar ingenieros al mercado, que catalicen y apalanquen las áreas de TI o fábricas de software a este modelo de operación y de esta manera aportar a la agilidad de los negocios.
- Extender la relación de correspondencia entre CMMI y metodologías ágiles en nuevos casos de estudio, abarcando más áreas de proceso CMMI que permitan obtener más datos y conseguir un mejor conocimiento de la materia en cuestión.
- Es posible la utilización del modelo propuesto en procesos de desarrollo rígidos dentro de empresas complejas, para esto es recomendable el cubrir las áreas que se excluyen en el modelo con prácticas de metodologías tradicionales o ágiles en función de los requerimientos tanto de proceso como de características de empresa.

- Es recomendable servirse de la utilización de herramientas informáticas como hojas de cálculo para facilitar el trabajo de gestión tanto la Tabla de Sprint como la Gráfica de Backlog, sin embargo la utilización de un papel impreso que se muestre en un lugar al cual el equipo de trabajo tenga acceso, tipo rotulador es más efectivo y simple de utilizar.
- Se recomienda evitar ciclos de desarrollo extensos mediante la especificación de Sprints “cortos”, esto favorece la disminución de errores en el desarrollo disminuyendo también los tiempos en los periodos de pruebas.
- Utilizar herramientas para metodologías ágiles que habiliten la implementación en empresas que tengan la iniciativa de respaldar la operación de área de TI con el uso de prácticas ágiles.
- Es recomendable mantener una política de capacitación y actualización constante en tecnologías de desarrollo y dispositivos informáticos relacionados con los proyectos abordados en el presente trabajo, puesto que el conocimiento y habilidad del equipo de desarrollo es un factor determinante en el éxito de un proyecto.

BIBLIOGRAFIA

- Software-Quality-Assurance.org. “CMMI Verification (VER) Process Area”. 2011. [Recuperado: 5-diciembre-2011]<http://www.software-quality-assurance.org/cmml-verification.html>
- JANE TOUMIKOSKI. “TESTING IN SCRUM”. 2009.[Recuperado: 20-noviembre-2011]
http://www.tol.oulu.fi/users/ilkka.tervonen/Ote_vierailu_09.pdf
- SMARTMATIX. “Scrum, RUP, XP....Assessed”.2007.[Recuperado: 10-noviembre-2011]
<http://www.smartmatix.com/Resources/CMMIexplained/ScrumRUPXPassested.aspx>
- Gerardo Fernández Escribano. “eXtreme Programming”. 2002. [Recuperado:15-diciembre 2011]
<http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQWdpbGVzL1Byb2dyYW1hY2lvdj9FeHRyZW1hLVhQLnBkZg%3D%3D&cidReset=true&cidReq=II0162004>
- Ronald Jeffries. “XProgramming.com”. 1999.[Recuperado:11-Noviembre 2011] <http://xprogramming.com>
- K. Beck. “Extreme Programming Explained: Embrace Change. Addison-Wesley, Pearson Education”. ISBN 201-61641-6. 2000.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides. “Design Patterns - Elements of Reusable Object Oriented Software. Addison-Wesley”, 1995.
- A. Herranz, J. Moreno, N. Maya. “Declarative reflection and its application as a pattern language.” 11th. Italy, 2002.

- A. Herranz and J. J. Moreno. "Generation of and debugging with logical pre and post conditions." Ducasse. Alemania. 2000.
- H.R. Berlack. "Software Configuration Management", John Wiley & Sons. 1992.
- Henrik Kniberg."Scrum y XP desde las Trincheras" .2012.[Recuperado:9- Octubre- 2011]. <http://www.proyectalis.com>

ANEXOS

Los anexos que se enumeran a continuación se encuentran en el disco compacto que acompaña a este documento.

Anexo1. HERRAMIENTAS RECOMENDADAS PARA INTEGRACION CONTINUA.

Anexo2. ACTA DE PLANIFICACIÓN DE PROYECTO.

Anexo3. DESARROLLO DE ITERACIONES EN SPRINTS.