Definición de Proceso Scrum



Trabajo presentado al profesor: César Jesús Pardo Calvache, PhD. MS.c. Eng.

Agile Solution

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Programa de Ingeniería de Sistemas
Asignatura Electiva Metodologías Ágiles
Popayán Noviembre de 2024

Indice	
Indice	2
Indice de Figuras	11
1. Resumen	12
2. Introducción	12
Figura 1.0 Logo del grupo Agile Solution	13
4. Objetivos	13
5. Cuerpo del trabajo	14
5.1. Proceso Scrum	14
Figura 1.1 Proceso Scrum	14
5.2. Inicio	15
Figura 1.2 Proceso Fase de Inicio.	16
Figura 1.3 Proceso Fase de inicio continuación	17
5.2.1. Elementos del proceso	18
5.2.1.1. Revisión del caso de negocio	18
5.2.1.2. Establecer la visión del proyecto	18
5.2.1.3. Identificar partes interesadas	18
5.2.1.4. Seleccionar scrum master	18
5.2.1.5. Identificar los miembros del equipo	19
5.2.1.6. Capacitación sobre Scrum	21
5.2.1.7. Establecer product backlog	22
Figura 2. Plantilla Product Backlog.	23
Figura 3. Tabla de abreviaturas .	23
Figura 4. Plantilla Elementos Generales del Product Backlog.	25
5.2.1.8. Establecer historias épica	25
Figura 5. Plantilla Historias Épicas	26
5.2.1.9. Registrar item en el product backlog	26
Figura 6. Visualización del Product Backlog actualizado	27
5.2.1.10. Creación de historias de usuario	27
Figura 7. Plantilla Historias De Usuario	27
5.2.1.11. Establecer criterios de aceptación	28
Figura 8. Plantilla Criterios de Aceptación	28
5.2.1.12. Establecer DoR	29
Figura 9. Plantilla DoR	29
5.2.1.13. Establecer DoD	30
Figura 10. Plantilla DoD	30
5.2.2. Recursos	31
5.2.2.1. Product Owner (Rol)	31
5.2.2.2. Scrum Master (Rol)	31
5.2.2.3. Scrum Team (Rol)	31
5.2.2.4. Notificación al product owner (Mensaje)	31
5.2.3. Subprocesos	31
5.2.3.1. Actualizar Product Backlog	31

5.2.3.1.1. Recibir product backlog actualizado

31

5.2.3.1.2. Guardar Product Backlog actualizado en un repositorio de Versiones	31
5.2.3.1.3. Relacionar elementos del product backlog acorde al objetivo del sprint	32
5.2.3.1.4. Organizar las funcionalidades en función del diagrama de paquetes	32
5.2.3.1.5. Organizar de mayor a menor prioridad el product backlog	32
5.2.3.1.6. Publicar Product Backlog	32
5.2.3.1.7. Notificar al scrum team la actualización	32
Figura 11. Proceso de actualización del product Backlog	33
5.2.3.2. Estimar Item	34
5.2.3.2.1. Recibir item	34
5.2.3.2.2. Establecer escala de estimación para priorizar	34
5.2.3.2.3. Establecer priorización de la historia de uso según escala	34
5.2.3.2.4. Establecer esfuerzo necesario para completar la historia según No Estimates	34
Figura 12. Proceso de estimación de ítems	35
5.3. Refinamiento del Product Backlog	36
Figura 13. Proceso de Refinamiento del Product Backlog	37
5.3.1. Elementos del proceso	38
5.3.1.1. Crear lista de registro de novedades	38
5.3.1.2. Revisar Product Backlog	38
Figura 14. Product Backlog	38
5.3.1.3. ¿Está el product backlog actualizado?	39
5.3.1.4. ¿Hubo algún cambio en el Sprint Product Backlog?	39
5.3.1.5. ¿Hay nuevos requisitos entrantes?	39
5.3.1.6. Registrar anotación del ingreso de nuevos requisitos	39
Figura 15. Registro de ingreso de nuevos requisitos al backlog	39
5.3.1.7. Examinar item del product backlog	39
5.3.1.8. ¿Se considera item obsoleto?	40
5.3.1.9. ¿El ítem presenta una estimación acorde a las necesidades del proyecto?	40
5.3.1.10. Registrar anotación de eliminación de item obsoleto.	40
Figura 16.Proceso de anotación por eliminación.	40
5.3.1.11. Evaluar siguiente ítem	40
5.3.1.12. Indicar progreso de trabajo en porcentaje del ítem	40
5.3.1.13. Evaluar criterio de hecho para verificar si el item esta completado	41
5.3.1.14. Registrar anotación de priorización de item no completado	41
5.3.2. Recursos	41
5.3.2.1. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1	41
5.3.2.2. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2	41
5.3.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3	41
5.3.2.4. Correo preguntando al Product Owner si hubo algún cambio en el product backlog	41
5.3.2.5. Esperar a que se presente algún cambio	41
5.3.2.6. Error en caso de no haber ningún cambio hasta el final del sprint	41
5.3.3. Subprocesos	42
5.3.3.1. Actualizar Product Backlog (Ya especificado en el inicio 4.1.3.1)	42
5.3.3.2. Convocar Reunión	42
Figura 17. Proceso de convocar reunión	43

5.3.3.2.1. Enviar Invitación de Reunión y Solicitud de Confirmación	44
5.3.3.2.2. Recepción de Respuestas de Asistencia	44
5.3.3.2.3. Notificar reprogramación de la reunión	44
5.3.3.2.4. Proponer nuevo horario	44
5.3.3.2.5. Confirmar reunión	44
5.3.3.2.6. Enviar notificación	44
5.3.3.3. Proceso de Gestión del Cambio	44
Figura 18. Proceso de planificación y estimación	45
5.4. Planificación y Estimación	46
5.4.1. Elementos del proceso	46
5.4.1.1. Solicitar reunión para planificación	46
5.4.1.2. Notificación de reunión	46
5.4.1.3. Verificar que las HU cumplan con el DoR	46
Figura 19. Proceso de revisión HU con DoR.	47
5.4.1.4. Notificación de reunión	48
5.4.1.5. inicio de reunión de planificación	48
5.4.1.6. Presentar y priorizar elementos del Product Backlog	48
5.4.1.7. Definir el Objetivo del Sprint	48
5.4.1.8. Estimar	49
5.4.1.9. Determinar cuántas historias de usuario se comprometen a realizar	51
5.4.1.10. ¿El trabajo estimado se ajusta a la capacidad del equipo?	51
5.4.1.11. Ajustar la selección de ítems	51
5.4.1.12. Crear un calendario de entregas.	51
5.4.1.13. Revisar y Ajustes finales	52
5.4.1.14. Aprobar el Sprint	52
5.4.2. Recursos	52
5.4.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2	52
5.4.2.2. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3	52
5.4.2.3. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1	52
5.4.2.4. Líder Técnico (Rol)	52
5.5. Implementación	52
Figura 20. Proceso de implementación.	53
5.5.1. Elementos del proceso	54
5.5.1.1. Daily Meeting	54
Figura 21. Proceso de Daily Meeting.	55
5.5.1.1.1. Programación de la reunión	56
5.5.1.1.2. Notificar reunión	56
5.5.1.1.3. Recibir notificación	57
5.5.1.1.4. Verificar disponibilidad	57
5.5.1.1.5. Enviar disponibilidad	57
5.5.1.1.6. Recibir disponibilidad	57
5.5.1.1.7. ¿Asistirá la mayoría del equipo?	57
5.5.1.1.8. Apertura de la reunión	58
5.5.1.1.9. Actualización del trabajo realizado	58

5.5.1.1.10. Identificación de impedimentos	58
5.5.1.1.11. Planificación del trabajo por hacer	59
5.5.1.1.12. Fin de la reunión	59
5.5.1.1.13. Planeación	59
5.5.1.1.14. Ejecución	60
5.5.1.1.15. Impediment Log	60
Figura 22. Plantilla del Impediment log.	60
5.5.1.2. Ejecución paralela	60
5.5.1.3. Gestión de Impedimentos	60
Figura 23. Proceso de gestión de impedimentos.	61
5.5.1.3.1. Recepción y revisión de impedimentos	61
5.5.1.3.2. Priorización de impedimentos	62
5.5.1.3.3. Búsqueda de soluciones	62
5.5.1.3.4. Ejecución	62
5.5.1.3.5. Actualización y seguimiento	63
5.5.1.3.6. Fin	63
5.5.1.3.7. Impediment Log	63
5.5.1.4. Desarrollo	63
Figura 24. Proceso de desarrollo.	64
5.5.1.4.1. Inicio del subproceso de desarrollo	65
5.5.1.4.2. Codificación y desarrollo de funcionalidades	65
5.5.1.4.3. ¿Codificación preparada para revisión?	65
5.5.1.4.4. Revision entre pares (Code review)	66
5.5.1.4.5. ¿El par aprobó los cambios?	66
5.5.1.4.6. Revisión del feedback	66
5.5.1.4.7. Inicio del subproceso pruebas unitarias e integración	67
5.5.1.4.8. Pruebas unitarias e integración	67
5.5.1.4.9. Finalización del subproceso pruebas unitarias e integración	67
5.5.1.4.10. ¿Pruebas unitarias exitosas?	67
5.5.1.4.11. Inicio del subproceso integración continua	67
5.5.1.4.12. Integración continua	68
5.5.1.4.13. ¿El subproceso Integración continua escaló?	68
5.5.1.4.14. Evaluación DoD	68
5.5.1.4.15. Documentación técnica	68
5.5.1.4.16. Fin del subproceso Desarrollo	68
5.5.1.4.17. Documentación de API y endpoints	69
Figura 25. Plantilla de documentación de Endpoints.	69
5.5.1.4.18. Registro de decisiones técnicas	69
Figura 26. Plantilla de Registro de decisiones técnicas.	70
5.5.1.5. Fin ejecución paralela	70
5.5.1.6. Evaluación DoD	70
Figura 25. Proceso Evaluación DoD	71
5.5.1.6.1. Elementos del proceso	72
5.5.1.6.2. Inicio de evaluación de DoD	72

	5.5.1.6.3. Notificación de revisión del Sprint	72
	5.5.1.6.4. Se aseguran que las funcionalidades desarrolladas cumplan con la DoD	72
	5.5.1.6.5. ¿Las funcionalidades cumplen con el DoD?	72
	5.5.1.6.6. Se listan las funcionalidades que no pasaron la DoD	73
	5.5.1.6.7. Se escala la desaprobación	73
	5.5.1.6.8. Se finaliza exitosamente	73
	5.5.1.6.9. Lista de funcionalidades que no pasaron la evaluación de DoD	73
	5.5.1.7. Pruebas unitarias y de integración	73
	Figura 27. Proceso de Pruebas unitarias e integración.	74
	5.5.1.7.1. Planificación de pruebas	75
	5.5.1.7.2. Desarrollo de pruebas unitarias	75
	5.5.1.7.3. Desarrollo de pruebas de integración	75
	5.5.1.7.4. Fin del subproceso de pruebas unitarias e integración	75
	5.5.1.7.5. Plan de pruebas	76
	5.5.1.7.6. Documentación técnica	76
	5.5.1.7.7. Documentación de flujos de integración	76
	5.5.1.8. Integración continua	76
	Figura 28. Proceso de integración continua	77
	5.5.1.8.1. ¿En qué fase está la célula?	78
	5.5.1.8.2. Compuerta Basada en Eventos	78
	5.5.1.8.3. Commit en la rama de desarrollo	78
	5.5.1.8.4. Pull request en la rama principal	79
	5.5.1.8.5. Construcción de builds	79
	5.5.1.8.6. Ejecución de pruebas automáticas	80
	5.5.1.8.7. Analizar resultados	81
	5.5.1.8.8. Resultado del análisis	82
	5.5.1.8.9. Despliegue a entorno de pruebas	82
	5.5.1.8.10. Se escala el subproceso	83
	5.5.1.8.11. Se escala el subproceso	83
	5.5.1.8.12. Configuración inicial	84
	5.5.1.8.13. Configuración de entorno de integración continua	84
	5.5.1.8.14. Creación de Builds automáticas	84
	5.5.1.8.15. Fin de la configuración	85
	5.5.1.8.16. Mantenimiento del Sistema de CI	85
	5.5.1.8.17. Revisión y ajuste periódico de la configuración del sistema de integración continua	a86
	5.5.1.8.18. Fin del mantenimiento	86
	5.5.1.8.19. Documentación de configuración, scripts de instalación	86
	5.5.1.8.20. Configuraciones de CI (e.g. Jenkins)	87
	5.5.1.8.21. Builds	87
	5.5.1.8.22. Informe de pruebas	87
	5.5.1.8.23. Incremento del Producto	87
5.5.	2. Recursos	88
	5.5.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2	88
	5.5.2.2. Equipo de desarrollo (Rol)	88

5.5.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3	88
5.5.2.4. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1	88
5.5.2.5. Stakeholders (Rol) Ya especificado en el inicio 4.8.2.4	88
5.6. Revisión del Sprint	88
Figura 29. Proceso de Revisión HU con DoD	89
5.6.1. Elementos del proceso	90
5.6.1.1. Revisión HU con DoD	90
5.6.1.1.1. Notificación de revisión del sprint	90
5.6.1.1.2. Se aseguran que las HU cumplan con el DOD	90
5.6.1.1.3. ¿Las HU cumplen con el DOD?	90
5.6.1.1.4. Se agrega a las HU completadas	90
5.6.1.1.5. Se agrega a las HU sin completar	90
5.6.1.1.6. HU sin finalizar	90
5.6.1.1.7. HU finalizadas	91
5.6.1.2. Revisión del Sprint	91
Figura 30. Proceso de Revisión del sprint.	92
5.6.1.2.1. Solicitar reunión para revisión del sprint	93
5.6.1.2.2. Notificación de reunión	93
5.6.1.2.3. Garantizar que las HU cumplan con el DoD	93
5.6.1.2.4. Notificación de reunión	93
5.6.1.2.5. Socializar HU del Sprint	93
5.6.1.2.6. Retroalimentar el Sprint	94
5.6.1.2.7. ¿Cumple con el criterio de DoD?	94
5.6.1.2.8. Se actualiza el sprint	94
5.6.1.2.9. Product Backlog Repriorizado	94
5.6.1.2.10. Gestión de cambio	94
5.6.1.2.11. Estimación	94
5.6.2. Recursos	94
5.6.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2	94
5.6.2.2. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3	94
5.6.2.3. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1	94
5.7. Lanzamiento (Demo)	95
Figura 31. Proceso de Lanzamiento (Demo).	96
5.7.1. Elementos del proceso	97
5.7.1.1. Lanzamiento	97
5.7.1.1.1 Inicio	97
5.7.1.1.2. Actividades de planeación	97
5.7.1.1.3. Programación de la reunión	97
5.7.1.1.4. Notificar reunión	98
5.7.1.1.5. Recibir notificación	98
5.7.1.1.6. Verificar disponibilidad	98
5.7.1.1.7. Enviar disponibilidad	98
5.7.1.1.8. Recibir disponibilidad	98
5.7.1.1.9. ¿Asistirá la mayoría de los interesados	99

5.7.1.1.10. Diseñar presentación	99
5.7.1.1.11. Todo listo para el evento	99
5.7.1.1.12. Hacer presentación de lanzamiento	100
5.7.1.1.13. Dar Feedback de la presentación	100
5.7.1.1.14. Registrar observaciones	101
5.7.1.1.15. Revisar el incremento	101
5.7.1.1.17. Actualizar Product backlog	102
5.7.1.1.18. Realizar documento de soporte para el lanzamiento	102
5.7.1.1.19. Realizar convocatoria de Planificación y estimación	103
5.7.1.1.20. Fin del evento	103
5.7.1.1.21. Planeación	103
5.7.1.1.22. Desarrollo del evento	104
5.7.1.1.23. Documento de Observaciones	104
Figura 32. Plantilla de Observaciones.	104
5.7.1.1.24. Registro de Solicitud del Cambio	104
Figura 33. Plantilla de Registro de solicitud del cambio.	105
5.7.1.1.25. Documento de Soporte de Lanzamiento	105
5.7.1.1.26. Validación de Lanzamiento	105
Figura 34. Plantilla de Validación de lanzamiento.	106
5.7.1.2. Diseñar Presentación	106
5.7.1.2.1. Inicio	106
5.7.1.2.2. Recopilar información del Sprint	107
5.7.1.2.3. Definir puntos claves	107
5.7.1.2.4. Creación de material visual	107
5.7.1.2.5. Preparación de la demostración del Incremento	108
5.7.1.2.6. Preparación técnica	108
5.7.1.2.7. Fin de diseñar presentación	108
5.7.2. Recursos	109
5.7.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2	109
5.7.2.2. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1	109
5.7.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3	109
5.7.2.4. Stakeholders (Rol) Ya especificado en el inicio 4.8.2.4	109
5.8. Retrospectiva del sprint	109
5.8.1. Elementos del proceso	109
5.8.1.1. Preparar materiales y espacio de trabajo	109
5.8.1.2. Iniciar retrospectiva del sprint	109
5.8.1.3. Indicar los objetivos y las actividades que se van a realizar	110
5.8.1.4. Confirmar asistencia	110
5.8.1.5. Dirigir actividad	111
5.8.1.6. Actividad de integración	111
5.8.1.7. Iniciar retrospectiva de Mad, Sad, Glad	112
5.8.1.8. Reflexión	112
5.8.1.9. ¿Qué salió bien?	113
5.8.1.10. Analizar comentarios	113
	113

5.8.1.11. Analizar de inconvenientes	113
5.8.1.12. Priorizar inconvenientes	114
5.8.1.13. Discusión	114
5.8.1.14. Consenso del equipo	115
5.8.1.15. Describir acciones de mejora	115
5.8.1.16. Establecer compromisos	116
5.8.1.17. Actualizar en la plantilla de retrospectiva del sprint	116
5.8.1.18. Redactar el plan de mejora y compromisos	117
5.8.1.19. Resumir retrospectiva	117
5.8.1.20. Concluir la reunión	118
5.8.1.21. ¿Qué salió mal?	118
5.8.1.22. ¿Qué aprendimos?	119
5.8.1.23. ¿Qué deberíamos hacer diferente la próxima vez?	119
5.8.1.24. Feedback de sentimientos	120
5.8.1.25. Plantilla retrospectiva del Sprint	120
5.8.1.26. Mad, sad, glad	120
5.8.1.27. Plan de mejora	121
5.8.1.28. Tableros	121
5.8.1.29. Backlog	121
5.8.1.30. Cronogramas	121
5.8.1.31. Paso 1	121
5.8.1.32. Paso 2	121
5.8.1.33. Paso 3	122
5.8.1.34. Paso 4	122
5.8.1.35. Paso 5	122
5.8.2. Recursos	122
5.8.2.1. Scrum Master (Rol)	122
5.8.2.2. Scrum Team (Rol)	122
5.9. Reunión de Gestión del Cambio	122
5.9.1. Elementos del proceso	123
5.9.1.1. ¿Lista de cambios aprobados está vacía ?	123
5.9.1.2. Refinamiento del Backlog	123
5.9.1.3. Comienzo del Proceso de Gestión del Cambio	123
5.9.1.4. Preparar materiales y espacio de trabajo	123
5.9.1.5. Iniciar reunión de gestión del cambio	124
5.9.1.6. Definición final de tareas	124
5.9.1.7. Planificación y Estimación	124
5.9.1.8. Finalización del Proceso de Gestión del Cambio	125
5.9.1.9. Lista de asistencia	125
5.9.1.10. Lista de cambios propuestos	125
5.9.1.11. Justificación de no aprobado	125
5.9.1.12. Lista de cambios aprobados	125
5.9.1.13. Lista de tareas refinadas	125
5.9.1.14. Descripción detallada de cada tarea	125

5.9.1.15. Product Backlog	125
5.9.1.16. Identificación posibles cambios potenciales	126
5.9.1.17. Socialización de los cambios	126
5.9.1.18. Iterar lista de posibles cambios propuestos	126
5.9.1.19. ¿La lista de cambios propuestos está vacía?	127
5.9.1.20. Seleccionar un cambio de la lista	127
5.9.1.21. Finalizar iteración de posibles cambios propuestos	127
5.9.1.22. ¿Cambio aprobado?	128
5.9.1.23. Evaluación del impacto	128
5.9.1.24. Justificar la desaprobación	128
5.9.1.25. Agregar a la lista de cambios aprobados	129
5.9.1.26. Identificación del tipo de cambio	129
5.9.2. Recursos	129
5.9.2.1. Product Owner (Rol)	129
5.9.2.2. Development Team (Rol)	130
5.9.2.3. Scrum Master (Rol)	130
5.9.2.4. Stakeholders (Rol)	130
6. Conclusiones	131
7. Bibliografía	132

Indice de Figuras

Figura 1.1 Proceso Fase de Inicio.	20
Figura 1.2 Proceso Fase de inicio continuación	21
Figura 2. Plantilla Product Backlog.	27
Figura 3.Tabla de abreviaturas.	27
Figura 4. Plantilla Elementos Generales del Product Backlog.	29
Figura 5. Plantilla Historias Épicas	30
Figura 6. Visualización del Product Backlog actualizado	31
Figura 7. Plantilla Historias De Usuario	31
Figura 8. Plantilla Criterios de Aceptación	32
Figura 9. Plantilla DoR	33
Figura 10.Plantilla DoD	34
Figura 11.Proceso de actualización del product Backlog	37
Figura 12.Proceso de estimación de ítems	39
Figura 13.Proceso de Refinamiento del Product Backlog	41
Figura 14. Product Backlog	42
Figura 15. Registro de ingreso de nuevos requisitos al backlog	43
Figura 16.Proceso de anotación por eliminación.	44
Figura 17. Proceso de convocar reunión	47
Figura 18. Proceso de planificación y estimación	49
Figura 19. Proceso de revisión HU con DoR.	51
Figura 20. Proceso de implementación.	57
Figura 21. Proceso de Daily Meeting.	59
Figura 22. Plantilla del Impediment log.	64
Figura 23. Proceso de gestión de impedimentos.	65
Figura 24. Proceso de desarrollo.	68
Figura 25. Plantilla de documentación de Endpoints.	73
Figura 26. Plantilla de Registro de decisiones técnicas.	74
Figura 27. Proceso de Pruebas unitarias e integración.	75
Figura 28. Proceso de integración continua.	78
Figura 29. Proceso de Revisión HU con DoD	88
Figura 30. Proceso de Revisión del sprint.	91
Figura 31. Proceso de Lanzamiento (Demo).	95
Figura 32. Plantilla de Observaciones.	103
Figura 33. Plantilla de Registro de solicitud del cambio.	104
Figura 34. Plantilla de Validación de lanzamiento.	104
Figura 35. Proceso de Diseñar Presentación.	106
Figura 36. Proceso de Retrospectiva del Sprint.	110
Figura 37. Proceso de Retrospectiva del Sprint Continuación.	111

1. Resumen

Este documento presenta un proceso ágil basado en Scrum, enfocado en la planificación, desarrollo y entrega iterativa de productos de software. El proceso se divide en fases claves, incluyendo Inicio, Refinamiento del Backlog, Planificación y Estimación, Revisión del Sprint, Retrospectiva del Sprint, Gestión del Cambio, Implementación y Lanzamiento.

Cada fase integra subprocesos específicos que facilitan la organización de tareas, priorización y validación continua de cada artefacto y pasos que se estén ejecutando. Además de especificar en cada momento que es lo que se hace para lograr hacer los subprocesos correctamente y que artefactos se usan en cada uno de estos.

2. Introducción

En el entorno dinámico y altamente competitivo del desarrollo de software, la necesidad de adoptar enfoques ágiles se ha vuelto esencial para garantizar entregas rápidas, eficientes y de alta calidad. Scrum, como uno de los marcos ágiles más utilizados, facilita la gestión de proyectos mediante ciclos iterativos, mejorando la colaboración entre equipos y permitiendo una adaptación continua a los cambios. Este documento describe un proceso ágil basado en Scrum que abarca desde la planificación hasta el lanzamiento de un producto, con el objetivo de optimizar la ejecución de cada sprint y mejorar la calidad de los entregables.

El proceso se organiza en fases claves: Inicio, donde se establece la visión del proyecto, se forma el equipo y se definen los criterios esenciales como el backlog, sus historias épicas y de usuario , también por supuesto sus respectivos criterios; Refinamiento del Backlog, que permite visualizar todos los cambios que hayan surgido en el product backlog y redactarlos para su fácil comprensión; Planificación y Estimación, donde el Scrum Team estime los ítems del product backlog y seleccione cuales harán parte del sprint backlog a desarrollar.; Revisión del Sprint, que permite evaluar los resultados entregados y ajustar los elementos del backlog según sea necesario; Retrospectiva del Sprint, para reflexionar sobre el proceso y aplicar mejoras; Gestión del Cambio, que asegura la correcta implementación de cambios en el proyecto; Implementación, en la que se construye el producto conforme a las prioridades establecidas; y Lanzamiento, que culmina con la demostración de los incrementos completados.

Cada una de estas fases está diseñada para garantizar un flujo de trabajo ágil y eficiente, promoviendo la mejora continua del proceso y el producto.

3. Agile Solution



Figura 1.0 Logo del grupo Agile Solution

Agile Solution es el grupo de estudiantes de la Electiva de Taller de Metodologías Ágiles que desarrolló un proceso basado en Scrum utilizando la plataforma Bizagi. Este proyecto surgió con el objetivo de detallar cada elemento de Scrum, incluyendo fases, roles, actividades y eventos, para crear un manual de procedimientos aplicable a micro, medianas y grandes empresas de desarrollo de software. La iniciativa respondió a la necesidad de contar con un proceso formal y documentado que facilite la implementación de Scrum en organizaciones con experiencia limitada en metodologías ágiles.

El trabajo buscó abordar una problemática común en el uso de enfoques ágiles: la falta de guías estructuradas que permitan su adopción efectiva. Aunque Scrum es ampliamente utilizado, no describe en detalle las tareas y actividades complementarias necesarias para su correcta aplicación. Por ello, Agile Solution diseñó un proceso completo y formal, adaptado al contexto regional, que proporciona herramientas claras para que las empresas puedan organizarse, ser más eficientes y promover la autoorganización de sus equipos de trabajo.

4. Objetivos

- Alinear a todos los participantes con la visión estratégica del proyecto estableciendo una base mediante la definición del caso de negocio, la identificación de los stakeholders y la selección del Scrum Master y los miembros del equipo, la eliminación de ítems obsoletos y la priorización ajustada a las necesidades actuales del proyecto
- Definir un proceso estandarizado basado en Scrum que permita a micro, pequeñas y medianas empresas de desarrollo de software aplicar prácticas ágiles de forma eficiente, detallando fases, actividades, roles, artefactos y eventos del proceso, con el fin de mejorar la organización, calidad y velocidad en la entrega de productos de software.

5. Cuerpo del trabajo

5.1. Proceso Scrum

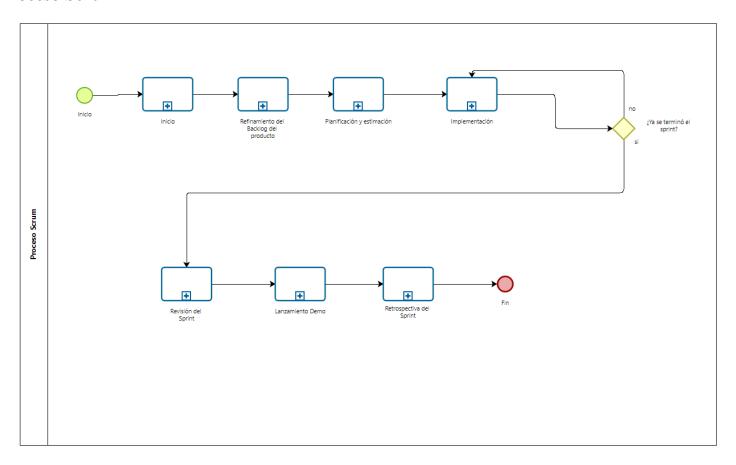


Figura 1.1 Proceso Scrum

Descripción

Scrum es un marco de trabajo ágil diseñado para gestionar proyectos de manera iterativa e incremental, promoviendo la colaboración, la adaptabilidad y la entrega continua de valor. Este enfoque se estructura en una serie de subprocesos clave que permiten a los equipos organizarse y avanzar de manera efectiva hacia los objetivos establecidos.

El proceso inicia con la fase de inicio, donde se definen los aspectos fundamentales del proyecto, como la visión del producto, el equipo Scrum y los objetivos iniciales. Esta fase establece una base sólida para el trabajo posterior y alinea a todos los participantes en torno a una meta común. Posteriormente, se lleva a cabo el refinamiento del Product Backlog, en el cual el Product Owner y el equipo colaboran para priorizar, detallar y dividir los elementos del backlog en tareas manejables. Este subproceso garantiza que el trabajo a realizar sea claro y relevante para los objetivos del sprint.

La planificación y estimación marca el comienzo formal de cada sprint, donde el equipo selecciona los elementos del Product Backlog que abordará y los descompone en tareas específicas. Durante esta etapa, se asignan prioridades, se estiman esfuerzos y se genera el Sprint Backlog, asegurando un enfoque claro y alcanzable para el período de trabajo. A continuación, en la fase de implementación, el equipo de desarrollo trabaja colaborativamente para completar las tareas asignadas, utilizando tableros o herramientas de seguimiento que promuevan la transparencia y la autoorganización.

Hacia el final del sprint, se realiza la revisión del sprint, una reunión donde el equipo presenta el incremento del producto desarrollado a los stakeholders. Este evento proporciona la oportunidad de recibir retroalimentación valiosa para ajustar el trabajo futuro según las necesidades del cliente o las circunstancias del proyecto. Después de la revisión, tiene lugar el lanzamiento (demo), donde el producto o incremento final es entregado formalmente y se evalúa su cumplimiento con los objetivos planteados.

Finalmente, el ciclo culmina con la retrospectiva del sprint, un espacio donde el equipo reflexiona sobre el desempeño del sprint. Aquí se identifican mejoras potenciales en los procesos, herramientas o dinámicas del equipo, buscando constantemente optimizar el trabajo y fortalecer la colaboración.

En conjunto, estos subprocesos hacen de Scrum un marco de trabajo robusto y adaptable, ideal para proyectos que requieren flexibilidad y un enfoque iterativo en la entrega de valor.

5.2. Inicio

Descripción

La fase de inicio es un conjunto de actividades previas al primer sprint que establece las bases para el desarrollo del proyecto. El propósito de esta fase es garantizar que todos los participantes (stakeholders, Product Owner, Scrum Master, y Scrum Team) estén alineados con la visión del producto, los objetivos del proyecto y los elementos fundamentales que guiarán el proceso de desarrollo ágil. Aunque Scrum no prescribe formalmente una fase de inicio, es una práctica recomendada para asegurar un comienzo sólido y organizado.

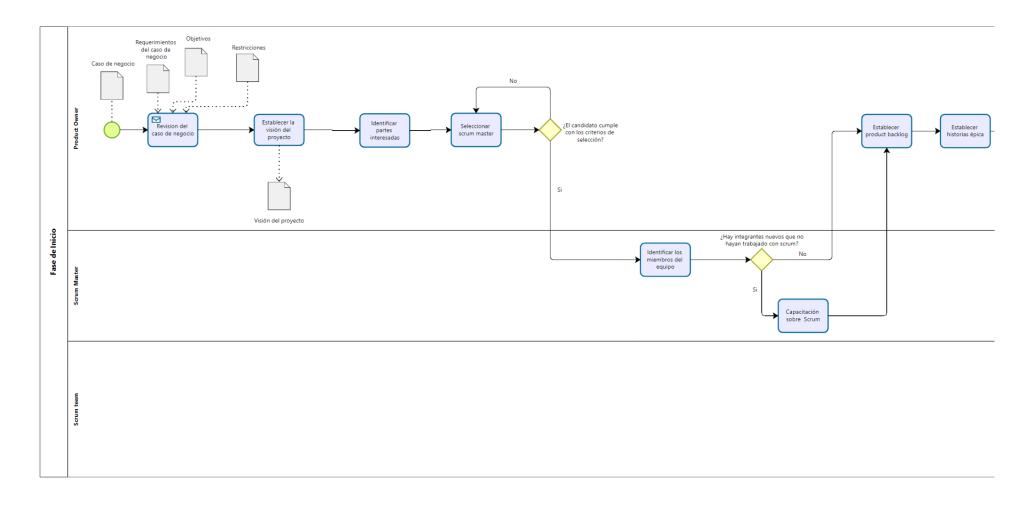


Figura 1.2 Proceso Fase de Inicio.

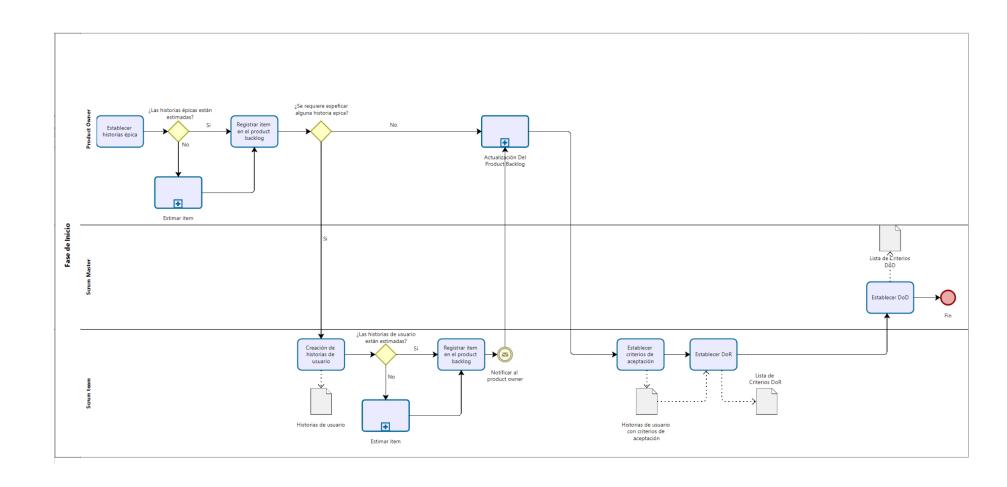


Figura 1.3 Proceso Fase de inicio continuación

5.2.1. Elementos del proceso

5.2.1.1. Revisión del caso de negocio

La revisión del caso de negocio permite evaluar y validar la viabilidad del caso de negocio para asegurar que los objetivos del proyecto estén alineados con las expectativas de los stakeholders y los objetivos estratégicos de la organización.

5.2.1.2. Establecer la visión del proyecto

El Establecimiento de la Visión del Proyecto es un paso fundamental al inicio de cualquier proyecto ágil, ya que proporciona una guía estratégica clara que orienta a todo el equipo de desarrollo y a los stakeholders en un objetivo común. Esta tarea asegura que el equipo entienda cuál es el propósito del proyecto, a quién está dirigido, qué problemas resuelve y cuál es el impacto esperado en la organización y los usuarios finales.

Esta fase no solo se enfoca en definir qué se quiere construir, sino también en establecer por qué es importante y cómo se medirá el éxito del proyecto.

El objetivo de esta tarea es crear un documento que describa de forma clara y concisa la visión del proyecto. Este documento servirá como un "norte" durante todo el ciclo de desarrollo, ayudando a alinear las expectativas de los stakeholders con el trabajo del equipo de desarrollo. Una visión bien definida es crucial para priorizar las funcionalidades y tomar decisiones de manera eficiente durante los sprints.

Roles Involucrados

Product Owner: Principal responsable de liderar la definición de la visión. Se encarga de recopilar la información necesaria de los stakeholders y traducirla en una declaración de visión comprensible y alcanzable.

Stakeholders: Proveen información clave sobre los problemas que el proyecto debe resolver, los objetivos comerciales, y las expectativas de valor para los usuarios.

Scrum Master: Facilita el proceso, asegurando que todas las partes interesadas participen y que la visión sea compartida y comprendida por el equipo.

Equipo de Desarrollo: Puede participar en sesiones de definición para ofrecer perspectivas técnicas y asegurar que la visión sea realizable.

La Identificación de las Partes Interesadas es una tarea clave que se lleva a cabo después de haber establecido la visión del proyecto. El objetivo de esta tarea es reconocer a todas las personas, grupos y entidades que tienen un interés, directo o indirecto, en el proyecto, así como en sus resultados. Esta actividad garantiza que se comprenden las expectativas, necesidades y niveles de influencia de cada stakeholder, lo que es crucial para gestionar el proyecto de forma efectiva y asegurar el éxito del mismo.

5.2.1.4. Seleccionar scrum master

La selección del scrum master es una de las partes más críticas en un proceso scrum, para hacer una selección que logre que sea ágil y que mantenga ágil nuestro proceso scrum se crean unos pasos que se deben tener en cuenta a la hora de seleccionar al scrum master, estos son:

- Paso 1 (Definir los requisitos del ROL): Es importante y esencial definir claramente los requisitos y responsabilidades del scrum master los cuales pueden ser:
 - Experiencia en proceso SCRUM o en algún proceso ágil.
 - Habilidades de facilitación y liderazgo de equipos.
 - Capacidades para resolver o eliminar problemas o impedimentos y resolver conflictos.
 - o Conocimiento técnico relevante para el desarrollo del proyecto.
 - Certificaciones relevantes en SCRUM o algún proceso Ágil.
- Paso 2 (Publicar la Vacante): Debemos de publicar que estamos buscando al scrum master, esta publicación se puede hacer internamente dentro de la gente del proyecto, y externa en el caso de tener la posibilidad de tiempos y costos
- Paso 3 (Evaluación de Candidatos): Debemos revisar todos los curriculum de los candidatos que se comunicaron al estar interesados en el puesto, en esta evaluación tendremos en cuenta:
 - o Experiencia en Scrum.
 - o Certificaciones.
 - Conocimiento en procesos Ágiles y experiencia.
 - Habilidades interpersonales.

En este paso se seleccionan a los mejores candidatos para el puesto del scrum que deberían ser entrevistados.

- Paso 4 (Entrevistas de Candidatos): se hacen entrevistas con los candidatos preseleccionados, se les pueden hacer preguntas como:
 - ¿Puedes describir tu experiencia trabajando como Scrum Master?
 - o ¿Cómo manejas los impedimentos que enfrenta el equipo?
 - ¿Qué estrategias utilizas para asegurar la colaboración dentro del equipo?
 - ¿Cómo facilitas las reuniones Scrum (Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective)?
 - o ¿Cómo manejas los conflictos dentro del equipo?
- Paso 5 (Evaluación Práctica): Se debería considerar realizar una prueba Práctica donde se pueda darle una situación hipotética y veremos que tan capacitado está para resolver problemas y comprender el nivel del conocimiento para su puesto.
- Paso 6 (Toma de Decisión): después de completar los anteriores pasos reúna a un equipo de selección donde mostraras los posibles candidatos finales según el mejor currículum, la evaluación y su entrevista, aquí se seleccionará al candidato final.
- Paso 7 (Oferta y Onboarding): Extiende una oferta al candidato seleccionado, si el candidato acepta planifica un proceso de onboarding(es un proceso de integración que las empresas utilizan para incorporar a nuevos empleados de manera efectiva) para integrarlo al equipo y familiarizarse con el proyecto, la organización y las herramientas a utilizar, en caso que no acepte se busca a una segunda persona o se vuelve al paso 3.
- Paso 8 (Tener en cuenta): Se debe tener en cuenta que debemos de hacer un monitoreo constante sobre el scrum master apenas empiece a trabajar, esto lo debemos hacer para saber el feedback que se crea y monitorear su desempeño.

Para identificar los miembros del equipo para un proyecto scrum ya que es fundamental para el desarrollo ágil, se debe tener el cuenta los siguientes pasos para hacer una buena elección:

- Paso 1 (Definir los Requisitos del Proyecto): Antes de identificar a los miembros del equipo, es fundamental entender las necesidades del proyecto:
 - Objetivos del Proyecto: Clarifica qué se espera lograr con el proyecto.
 - o Tecnologías y Herramientas: Determina las tecnologías y herramientas que se utilizarán.
 - Habilidades Necesarias: Identifica las habilidades técnicas y no técnicas necesarias (desarrollo, diseño, pruebas, etc.).
- Paso 2 (Establecer el Tamaño del Equipo): El equipo Scrum debe ser lo suficientemente pequeño para mantenerse ágil, pero lo suficientemente grande para completar el trabajo. Generalmente, se recomienda que el equipo tenga entre 3 y 9 miembros.
- Paso 3 (Evaluar la Disponibilidad de Recursos Internos):Revisa los recursos disponibles dentro de la organización:
 - Habilidades Actuales: Evalúa las habilidades de los empleados actuales para determinar si cumplen con los requisitos del proyecto.
 - O Disponibilidad: Verifica la disponibilidad de los empleados y su capacidad para comprometerse con el proyecto.
- Paso 4 (Seleccionar a los Miembros del Equipo)
 - Desarrolladores: Busque desarrolladores con las habilidades técnicas necesarias (programación, diseño de bases de datos, etc.). Asegúrese de que sean capaces de trabajar de manera colaborativa y ágil.
 - Diseñadores: Incluya diseñadores de UX/UI si el proyecto requiere diseño de interfaces de usuario.
 - QA/Testers: Los testers son esenciales para asegurar la calidad del producto. Busque personas con experiencia en pruebas automatizadas y manuales.
 - Especialistas en DevOps: Si el proyecto requiere despliegue continuo y automatización, considere incluir especialistas en DevOps.
- Paso 5 (Evaluar candidatos): Realizar entrevistas y evaluaciones técnicas para asegurarte de que los candidatos seleccionados tienen las habilidades necesarias y son adecuados para trabajar en un entorno Scrum. Algunas preguntas clave pueden incluir:
 - ¿Cuál es tu experiencia trabajando en equipos ágiles y/o Scrum?
 - o ¿Cómo manejas los plazos ajustados y las prioridades cambiantes?
 - ¿Puedes dar un ejemplo de cómo has colaborado con otros miembros del equipo para resolver un problema técnico?
- Paso 6 (Formación del Equipo): Una vez seleccionados los miembros del equipo, asegúrese de que entiendan sus roles y responsabilidades dentro del equipo Scrum. Facilite una sesión de orientación para presentarles:
 - o La visión del proyecto
 - Objetivos del sprint
 - Procesos y ceremonias de Scrum (Daily Scrum, Sprint Planning, etc.)
- Paso 7 (Promover la Colaboración y la Cohesión del Equipo):Fomenta un ambiente de trabajo colaborativo desde el principio:
 - Team Building: Organice actividades de team building para fortalecer las relaciones entre los miembros del equipo.
 - o Comunicación Abierta: Promueve la comunicación abierta y la transparencia.
- Paso 8 (Asignar Roles Claros): Asegúrese de que cada miembro del equipo entienda su rol específico y cómo contribuye al objetivo del proyecto:

- Desarrolladores: Enfocados en la creación del producto.
- Testers: Garantizan la calidad del producto.
- Diseñadores: Trabajan en la experiencia del usuario y la interfaz.
- O DevOps: Manejan la infraestructura y el despliegue.
- Paso 9 (Establecer Expectativas Claras):Desde el principio, establece expectativas claras en cuanto a la participación en las ceremonias de Scrum, la entrega de trabajo y la colaboración en equipo.

5.2.1.6. Capacitación sobre Scrum

Para hacer una capacitación sobre SCRUM ya teniendo un equipo con el que se va a trabajar es importante para tener las ideas claras, la información fresca sobre cómo se trabajará en el proyecto, y cuales son sus roles, líderes y en caso de algún problema con quién comunicarse, para esto podríamos seguir los siguientes pasos:

- Paso 1 (Definir los Objetivos de la Capacitación): Antes de comenzar, es importante definir los objetivos específicos de la capacitación:
 - Asegurar que todos los miembros del equipo comprendan los principios y valores de Scrum.
 - o Enseñar las prácticas y ceremonias de Scrum.
 - o Clarificar los roles y responsabilidades dentro de un equipo Scrum.
 - Proveer habilidades prácticas para aplicar Scrum en proyectos reales.
- Paso 2 (Preparar el Material de Capacitación): Reúna y prepare todo el material necesario para la capacitación, que puede incluir:
 - o Presentaciones de diapositivas.
 - Manuales y guías de Scrum.
 - o Ejercicios prácticos y estudios de caso.
 - Recursos de referencia, como el Scrum Guide (Guía de Scrum).
- Paso 3 (Planificar la Agenda de Capacitación): Organice una agenda estructurada para cubrir todos los aspectos importantes de Scrum:
 - Día 1: Introducción a Scrum
 - Historia y origen de Scrum: Explique cómo y por qué se desarrolló Scrum.
 - Principios y valores de Scrum: Enfóquese en los principios de transparencia, inspección y adaptación, y los valores de coraje, foco, compromiso, respeto y apertura.
 - Comparación con otros marcos ágiles: Breve comparación con otros marcos ágiles como Kanban y XP.
 - o Día 2: Roles en Scrum
 - Scrum Master: Responsabilidades y habilidades necesarias.
 - Product Owner: Responsabilidades y habilidades necesarias.
 - Scrum Team: Composición y roles dentro del equipo de desarrollo.
 - Día 3: Artefactos de Scrum
 - Product Backlog: Creación, gestión y priorización.
 - Sprint Backlog: Selección y planificación del sprint.
 - Incremento: Definición y criterios de aceptación.
 - o Día 4: Ceremonias de Scrum
 - Sprint Planning: Objetivos y procesos.
 - Daily Scrum: Propósito y formato.
 - Sprint Review: Cómo presentar el incremento y obtener feedback.
 - Sprint Retrospective: Identificación de mejoras y acciones.
 - Día 5: Ejercicios Prácticos y Estudio de Casos
 - Ejercicio de planificación de sprint: Simulación de una planificación de sprint real.

- Daily Scrum simulada: Práctica de reuniones diarias.
- Revisión y retrospectiva de sprint: Simulación de una revisión y retrospectiva de sprint.
- Paso 4 (Seleccionar a los Facilitadores): Asegúrese de que los facilitadores de la capacitación sean personas con experiencia práctica en Scrum, idealmente certificadas como Scrum Master (CSM) o Professional Scrum Master (PSM).
- Paso 5 (Realizar la Capacitación): Lleve a cabo la capacitación según la agenda planificada:
 - Introducción y Expectativas: Comience cada sesión con una breve introducción y establezca las expectativas para el día.
 - Presentaciones y Discusiones: Utilice presentaciones para explicar los conceptos y fomente discusiones para aclarar dudas.
 - Ejercicios Prácticos: Incluya actividades prácticas para reforzar el aprendizaje teórico.
 - Estudios de Caso: Analice estudios de caso reales para mostrar cómo se aplica Scrum en diferentes situaciones.
 - Feedback y Ajustes: Al final de cada sesión, recoja feedback de los participantes para ajustar y mejorar las futuras sesiones.
- Paso 6 (Proveer Recursos de Seguimiento): Después de la capacitación, proporcione a los participantes recursos adicionales para continuar su aprendizaje:
 - Libros y artículos sobre Scrum: Recomendaciones de lecturas adicionales.
 - Acceso a comunidades de práctica: Foros, grupos de discusión y eventos sobre Scrum.
 - o Cursos en línea y webinars: Opciones para profundizar en temas específicos.
- Paso 7 (Evaluar el Impacto de la Capacitación):Después de completar la capacitación, evalúe su efectividad:
 - o Encuestas de Satisfacción: Recoja opiniones sobre la calidad de la capacitación.
 - Pruebas de Conocimiento: Realice pruebas para evaluar el entendimiento de los conceptos de Scrum.
 - Seguimiento en el Lugar de Trabajo: Observe la aplicación práctica de Scrum en los proyectos y proporciona retroalimentación continua.

5.2.1.7. Establecer product backlog

Después de capacitar al equipo en caso de ser necesario el product owner procede con la creación del Producto Backlog como estructura para después poder llenarlo teniendo en cuenta una plantilla previamente diseñada de la siguiente manera:

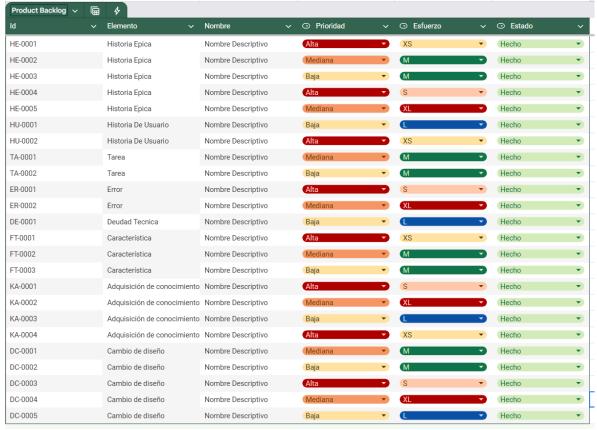


Figura 2. Plantilla Product Backlog.

Se tendrá en cuenta los siguientes aspectos para la creación del product backlog.

Elementos del Product Backlog:

- **Id:** Identificador del elemento con su respectiva abreviatura y numeración. Se recomienda las siguientes abreviaturas a la hora de colocar cada elemento representado en la siguiente tabla:



Figura 3. Tabla de abreviaturas.

- **Elemento:** En este caso, cada elemento se nombra de manera específica. Por ejemplo, si hay 'HU', se refiere a 'Historia de Usuario', con el fin de facilitar su comprensión al visualizar el Product Backlog.
- **Nombre:**En este espacio se asignará un nombre que describa de manera concisa el elemento, sin llegar a ser una descripción detallada
- Prioridad:La priorización es el proceso de añadir un valor a una tarea, error, característica o escenario y luego ordenar de lo más a lo menos importante en función a ese valor asignado[7].Por consiguiente hay diferentes maneras de estimar la prioridad de un elemento tenemos por ejemplo: "Marco RICE,Modelo Kano,Método MosCoW,Valor frente a efecto,Puntuación de oportunidades, Coste del retraso"[8]. En nuestro caso colocamos en la plantilla una priorización con escala de alta,mediana y baja teniendo en cuenta el valor que ese elemento le ofrezca al cliente.
- Esfuerzo: El esfuerzo en un product backlog es una estimación de la cantidad de trabajo que se requiere para completar cada elemento como en la priorización este tiene distintas formas y marcos o filosofías para poder definirlo en este documento nosotros utilizaremos NoEstimates que es un filosofía en el desarrollo ágil que hace referencia a la entrega pronta de valor al cliente en lugar de estimar de antemano y por eso dejando de lado las escalas numéricas y calculadas ,nosotros propusimos el tema de Tallas de camisetas que es una escala desde XS a XL para poder decir que tanto esfuerzo va a ser necesario para completar esa tarea.
- **Estado:**Identifica los posibles estados de la historia durante su ciclo de vida[1].Nosotros colocamos los estados de vacío,planificada,en proceso y hecho. Vacío:Cuando está definida pero no estimada;Planificada:Cuando está definida y estimada;En Proceso: Cuando esta siendo desarollada; Hecho:Cuando ya cumplio con todos los criterios que le corresponden.

Nota:En nuestro proceso no tenemos en cuenta el estado pero es importante mencionarlo para nuestra investigación.

Por otro lado están los tipos de Elementos que hay dentro de un Product Backlog, tomando referencia se obtuvo lo siguiente "Todos los elementos de trabajo deben incluirse en el backlog: historias de usuario, errores, cambios de diseño, deuda técnica, solicitudes de clientes, elementos de acción de la retrospectiva, etc.[4]" y "Usualmente este listado comprende diferentes tipos de elementos:Features,Bugs,Technical work, Knowledge acquisition[2]" de donde se destacan estos tipos:

- **Historia Épica**: Es una historia de usuario que por su gran tamaño generalmente el equipo descompone en historias con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles[9].
- **Historia De Usuario**: Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final[6]. Para definirla tenemos un estructura que se mencionara más adelante en el proceso de creación de historia de usuario.
- **Tarea**: Para producir una historia, el equipo necesita realizar actividades estructuradas como Tareas; Teniendo en cuenta que una tarea no es parte del resultado del proyecto, es más bien el medio para producir el resultado[10].
- Error:
- Deuda Técnica: La deuda técnica en el product backlog se coloca para visibilizar y priorizar aspectos técnicos pendientes que, de no atenderse, pueden afectar la calidad, rendimiento, o capacidad de mantenimiento del producto a largo plazo. La deuda técnica suele ser el resultado de decisiones como desarrollos rápidos, soluciones provisionales, o falta de documentación adecuada, que si no se resuelven, pueden acumularse y convertirse en problemas complejos.
- Característica: Las características representan partes del producto que aportan un valor significativo a sus usuarios[10]. Estas se pueden tardar en desarrollar hasta varios sprint.

- Adquisición De Conocimiento: Como lo dice su nombre es la parte que se coloca en el product backlog para especificar que se necesita aprender algo para poder continuar o empezar con el desarrollo.
- Cambio De Diseño: Aquí se especifican los cambios que surjan normalmente en la parte estética de un desarrollo, describiendo normalmente en qué interfaz sucedió y los cambios que puedan surgir de este cambio de diseño.

Nota: Como en el anterior punto en este no se abarcan todos en el proceso. Tome los de Historias de Usuario y Historias Épicas; También es importante recalcar que aunque los otros elementos no son tan generales como las historias épicas y de usuario también deben de estimarse y crearles sus criterios de aceptación.

Por consiguiente se generó una plantilla para futuras implementaciones y se tengan en cuenta esos elementos nombrados anteriormente.

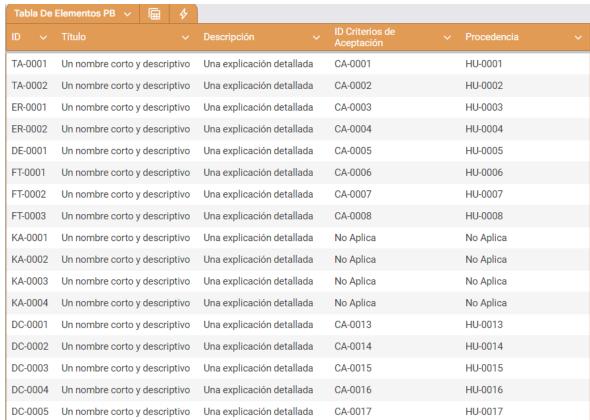


Figura 4. Plantilla Elementos Generales del Product Backlog.

Al igual que en plantillas anteriores, se cuenta con un ID para identificar de forma abreviada el tipo de elemento, un título que describe de manera concisa el contenido del elemento, los IDs de los criterios de aceptación que le corresponden y, por supuesto, la precedencia de donde proviene dicho elemento. Cabe aclarar que, en algunos casos, los criterios de aceptación y la precedencia pueden no ser aplicables; en esos casos, se indicará cómo 'No Aplica'.

5.2.1.8. Establecer historias épica

Después de haber creado la estructura del product backlog el product owner como se mencionaba antes, procederá a la creación de las primeras historias épicas, también teniendo en cuenta una estructura, elementos y una plantilla previamente definidos.



Figura 5. Plantilla Historias Épicas

Primero, es fundamental tener en cuenta que, para crear historias épicas, se deben basar en la visión previamente establecida del proyecto. Esto garantiza que las historias estén alineadas con el objetivo final del proyecto. Generalmente, se comienza definiendo lo que se quiere lograr con esa historia épica y se plasma en su descripción. Luego, se le asigna su respectivo ID y, al igual que con otros elementos, se le asigna el esfuerzo y la prioridad durante el subproceso de estimación.

5.2.1.9. Registrar item en el product backlog

Una vez todas las historias épicas o historias de usuario registradas las colocan en la cima de la pila del product backlog a medida que se registran. El Product Owner de vigilar como responsable de estos primeros procesos que las historias deban estar previamente estimadas en el subproceso de estimación de ítems. Luego, se procede al proceso de actualizar el product backlog, en el cual se establece el orden, teniendo en cuenta la priorización, el esfuerzo, las categorías ó su relación con otras historias y esto se vería de la siguiente manera:



Figura 6. Visualización del Product Backlog actualizado

5.2.1.10. Creación de historias de usuario

El Scrum Team, después de preguntarse si alguna historia épica necesita especificación, utilizará la estructura o plantilla destinada para la creación de historias de usuario. Sin embargo, antes de esto, es importante considerar un aspecto clave para crear una historia de usuario. El propósito de una historia de usuario es articular cómo un elemento de trabajo entregará un valor particular al cliente. Cabe recordar que los 'clientes' no necesariamente son usuarios finales externos; también pueden ser clientes internos o colegas dentro de la organización que dependen del equipo[6].

Por lo tanto, y como se mencionó anteriormente, aquí se desglosará un poco más la historia épica en cuantas historias de usuario sean necesarias. Si se derivan demasiadas historias de una sola épica, esto puede ser una señal de que la historia épica está mal diseñada y puede requerir ajustes.

Siguiendo con el tema principal, la historia de usuario también puede desglosarse en tareas y/o provenir de una característica ('feature') o generar una, lo cual aporta valor al cliente. Para la creación de la historia de usuario, utilizaremos la siguiente plantilla.

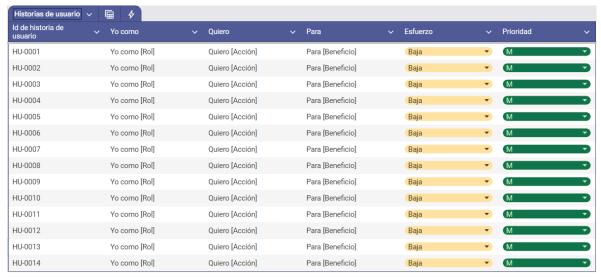


Figura 7. Plantilla Historias De Usuario

Teniendo en cuenta la [FIgura 7] podemos empezar a plantear cada uno de sus ítems

- **ID**:para identificar de forma abreviada el tipo de elemento
- Yo como: Aquí principalmente definiremos quien ejecuta o utilizara la función ,pero no solo eso pongamoslo de la siguiente manera primero seleccionamos el rol del usuario que vaya a

- ejecutar la acción y después colocamos alguna característica si es que la posee dentro del sistema esto teniendo en cuenta al usuario como centro de esa función. Ejemplo: Como un cliente que realiza compras frecuentes.
- Quiero: Describe la acción concreta que el usuario desea llevar a cabo. Esta debe de ser clara, concisa y enfocada en el objetivo.esto obviamente especificando el paso necesario para llevar a cabo esta acción.sin olvidar expresar la necesidad subyacente que motiva al usuario a realizar esta acción. Ejemplo: Quiero filtrar los resultados de búsqueda por categoría.
- **Para:** es el beneficio o resultado que el usuario obtiene al realizar la acción. Ejemplo: pueda encontrar más fácilmente los productos que busco.
- **Esfuerzo y Prioridad:** Estos también los tendremos en cuenta en esta plantilla y por supuesto deben de estar especificados antes en el subproceso de estimar item.

Una vez tengan las historias de usuario e historias épicas definidas y estimadas se **notificará al product owner** con las características definidas en el punto 4.1.2.4 para después en el siguiente paso proceder a redactar y divulgar sus criterios de aceptación.

5.2.1.11. Establecer criterios de aceptación

Los criterios de aceptación son condiciones específicas que una historia de usuario o tarea debe cumplir para ser aceptada como completa. Estos criterios son definidos por el Product Owner en colaboración con el equipo de desarrollo y los Stakeholders, y ayudan a establecer una comprensión compartida sobre los requisitos mínimos de una funcionalidad o ítem del backlog. Al tener criterios de aceptación bien definidos, el equipo puede asegurarse de que la implementación cumple con las expectativas y requisitos del negocio, reduciendo ambigüedades y mejorando la calidad del resultado final.

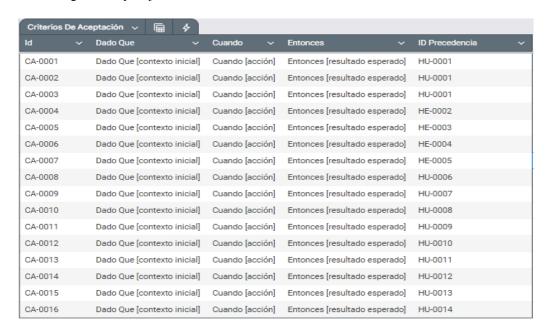


Figura 8. Plantilla Criterios de Aceptación

Id: Este es el identificador único del criterio de aceptación. Es un código que permite diferenciar cada criterio de aceptación de los demás, y facilita su seguimiento y referencia en el proyecto. Por ejemplo, "CA-0001" indica el primer criterio de aceptación.

Dado Que: Esta sección define el contexto o estado inicial en el que se encuentra el usuario o el sistema antes de realizar una acción. Es una forma de establecer las condiciones previas necesarias para que se desencadene el criterio de aceptación. En un contexto de desarrollo ágil, se utiliza para describir la situación inicial que activa el flujo de trabajo.

Cuando: Esta parte describe la acción o evento específico que el usuario o el sistema ejecuta en el contexto definido en "Dado Que". Es el desencadenante que produce una respuesta o resultado esperado.

Entonces: Esta sección define el resultado esperado o la respuesta del sistema una vez que se ha realizado la acción especificada en "Cuando". Este resultado es lo que se debe verificar para asegurar que el criterio de aceptación se ha cumplido.

ID Precedencia: Este campo indica la relación de dependencia con otras historias de usuario (HU) o elementos del backlog. Aquí se muestra el identificador de una historia de usuario de la cual depende este criterio de aceptación. Por ejemplo, "HU-0001" indica que el criterio depende de la historia de usuario 0001, y esta dependencia debe completarse o estar lista antes de que el criterio pueda ser verificado.

5.2.1.12. Establecer DoR

La Definición de Listo (DoR) es una lista de criterios que un ítem del backlog debe cumplir antes de ser considerado listo para que el equipo de desarrollo comience a trabajar en él. Esta definición incluye requisitos mínimos como tener una descripción clara, contar con criterios de aceptación bien definidos y resolver cualquier dependencia externa que pueda bloquear el desarrollo. La DoR garantiza que el equipo tenga toda la información necesaria para iniciar el trabajo, evitando retrasos o problemas por falta de claridad y asegurando un flujo de trabajo más eficiente y continuo.



Figura 9. Plantilla DoR

ID: Es el identificador único de cada criterio en la Definición de Listo (DoR). Este código permite distinguir cada criterio individualmente, lo cual facilita su rastreo y referencia en el proyecto. Por ejemplo, "DR-0001" identifica un criterio específico de la DoR.

Descripción: Esta sección contiene una explicación detallada de lo que se requiere para que el elemento se considere listo para empezar a trabajar. La descripción define las condiciones previas o los requisitos que deben cumplirse antes de que el equipo de desarrollo pueda abordar una historia de usuario o tarea.

Estado: El estado indica si el criterio en cuestión ha sido cumplido o no. Normalmente, se utiliza "Listo" cuando se han satisfecho todas las condiciones necesarias y "No Listo" cuando aún falta información o algún requisito por cumplir. Esto permite que el equipo vea claramente qué ítems están listos para comenzar y cuáles requieren más preparación.

Precedencia: Este campo muestra las dependencias que el criterio puede tener con otros elementos. Indica si el criterio de la DoR depende de una historia de usuario, tarea o actividad previa que debe completarse primero. Esto ayuda a asegurar que no se empiece a trabajar en algo que dependa de un ítem no completado y que podría bloquear el avance.

5.2.1.13. Establecer DoD

La Definición de Hecho (DoD) es un conjunto de criterios que un ítem debe cumplir al completarse para ser considerado terminado y listo para entregarse. Estos criterios suelen incluir estándares de calidad, pruebas satisfactorias, revisión de código y documentación. La DoD establece un marco común de calidad, asegurando que todo lo que el equipo entregue esté completamente terminado y listo para ser utilizado o integrado. Con la DoD, el equipo puede mantener un alto estándar en cada entrega, lo que contribuye a la satisfacción del cliente y la efectividad del proceso de desarrollo.



Figura 10. Plantilla DoD

ID: Este es el identificador único de cada criterio de la Definición de Hecho (DoD). Cada ítem de DoD tiene un código único que facilita su seguimiento, referencia y control. Por ejemplo, "DD-0001" representa un criterio específico de DoD.

Descripción: Este campo describe lo que se debe evaluar o cumplir para que el ítem se considere como "hecho". La descripción define claramente los requisitos o verificaciones necesarias para asegurar que el trabajo cumple con los estándares de calidad y requisitos especificados en la DoD. En esta sección se incluye el detalle de lo que cada elemento debe lograr para cumplir con el estándar de "terminado".

Estado: El estado indica si el criterio en cuestión ha sido cumplido o no. Normalmente, se utiliza "Hecho" cuando el criterio ha sido completado y validado, y "No Hecho" cuando el criterio aún no se ha cumplido. Esto permite un seguimiento visual del progreso y facilita la identificación de criterios pendientes o cumplidos.

Precedencia: Este campo muestra la relación de dependencia con otros elementos o tareas. En él se detalla si el cumplimiento del criterio depende de otro ítem (como una Historia de Usuario o Tarea) que debe completarse antes. Esto asegura que las dependencias necesarias estén resueltas y que no haya obstáculos que impidan que el criterio de DoD sea marcado como completado.

5.2.2. Recursos

5.2.2.1. Product Owner (Rol)

El Product Owner es el representante del cliente y la voz principal del negocio dentro del equipo Scrum. Su principal responsabilidad es maximizar el valor del producto que se está desarrollando, asegurando que el equipo Scrum trabaja en las funcionalidades correctas, priorizando los elementos que aporten mayor valor y alineando el trabajo con los objetivos del negocio.

5.2.2.2. Scrum Master (Rol)

El Scrum Master es un facilitador clave en todo el proceso de Scrum, actuando como un "guía" y un "protector" del equipo y de los principios de Scrum. Su rol se centra en ayudar al equipo a maximizar su eficiencia y en asegurar que Scrum se implemente correctamente.

5.2.2.3. Scrum Team (Rol)

El Scrum Team es un equipo ágil autoorganizado y multifuncional, compuesto por personas con diferentes habilidades y roles, que trabajan juntas para desarrollar y entregar incrementos de valor en cada Sprint.

5.2.2.4. Notificación al product owner (Mensaje)

Una vez se tengan las historias de usuario que se necesitaron para especificar las historias épicas que lo necesitaban un miembro del scrum team procede a notificar al product owner por medio de un email para que esté atento a cómo estas se desarrollaron y en caso de algún cambio sepa las afectaciones y sus riesgos; se hizo para enviar ya cuando esten todas las primeras historias de usuario para evitar estar mandando un mensaje cada que se especifique una.

5.2.3. Subprocesos

5.2.3.1.1. Recibir product backlog actualizado

En esta sección se recibe como entrada el product backlog actualizado a la fecha para poder actualizar todos los cambios que se tienen nuevos en el backlog.

5.2.3.1.2. Guardar Product Backlog actualizado en un repositorio de Versiones

Por supuesto una vez se tenga el product backlog actual hay que guardarlo en la carpeta de versiones para poder tener un control de versiones y retroceder si en algún momento se requiera ó revisar algún ítem que no debió ser modificado.

5.2.3.1.3. Relacionar elementos del product backlog acorde al objetivo del sprint

Aparte de él organizarlos por su prioridad también hay que buscar y establecer la relación entre cada uno de los ítems del product backlog esto para poder también facilitar el hacer el sprint product backlog.

5.2.3.1.4. Organizar las funcionalidades en función del diagrama de paquetes

Esta tarea permite estructurar las funcionalidades del sistema de manera lógica y coherente. Esta organización se basa en el uso de un diagrama de paquetes, el cual permite agrupar y categorizar los elementos del backlog según sus áreas funcionales, responsabilidades y dependencias dentro del proyecto.

5.2.3.1.5. Organizar de mayor a menor prioridad el product backlog

En este paso luego de agruparlos y corregir las escalas si es que era necesario procedemos a ordenarlos según la escala de priorización escogida.

5.2.3.1.6. Publicar Product Backlog

Una vez se haya verificado cada ítem se puede pasar a publicar el nuevo product backlog o product backlog actualizado y que todos puedan estar atentos a esos cambios en la pila.

5.2.3.1.7. Notificar al scrum team la actualización

Por último cuando se haya publicado el product backlog se le notifica al scrum team por medio de correo electrónico como se mencionó anteriormente esto para que estén atentos a los cambios y puedan en futuras revisiones opinar de una manera efectiva.

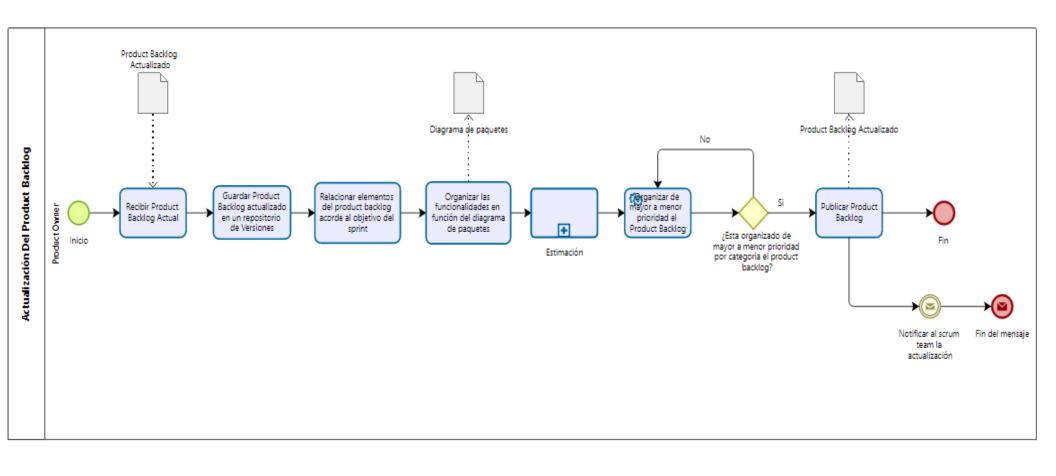


Figura 11. Proceso de actualización del product Backlog

5.2.3.2. Estimar Item

5.2.3.2.1. Recibir item

- Descripción:
 - El Product Owner proporciona el ítem del Product Backlog a ser estimado.
 - Se asegura de que el ítem esté bien definido y que tenga criterios de aceptación claros.
- Acciones:
 - o Product Owner: Presenta el ítem y explica los detalles y los criterios de aceptación.
 - o Equipo Scrum: Escucha y revisa el ítem, haciendo preguntas para aclarar cualquier duda.

5.2.3.2.2. Establecer escala de estimación para priorizar

- Descripción:
 - Acordar una escala de estimación que el equipo utilizará para todos los ítems del backlog.
- Acciones:
 - Equipo Scrum: Acordar utilizar una escala específica (por ejemplo, la secuencia de Fibonacci: 1, 2, 3, 5, 8, 13, 21).

5.2.3.2.3. Establecer priorización de la historia de uso según escala

- Descripción:
 - Utilizar la escala seleccionada para priorizar las historias de usuario en el backlog.
- Acciones:
 - Product Owner: Prioriza las historias de usuario basándose en el valor de negocio y la escala acordada.
 - o Equipo Scrum: Proporciona feedback y ayuda a ajustar las prioridades según sea necesario.

5.2.3.2.4. Establecer esfuerzo necesario para completar la historia según No Estimates

- Descripción:
 - Enfoque "No Estimates" se basa en priorizar y trabajar en ítems sin realizar estimaciones numéricas tradicionales. En su lugar, el equipo se enfoca en el flujo continuo y en la entrega rápida de valor.
- Acciones:
 - Equipo Scrum: Revisa el ítem y discute el esfuerzo necesario para completarlo, basándose en la experiencia previa y en la complejidad relativa.
 - Scrum Master: Facilita la discusión para asegurar que todos los miembros del equipo estén de acuerdo en la complejidad del ítem.
 - Product Owner: Asegura que la priorización refleja el valor de negocio y la urgencia de los ítems.

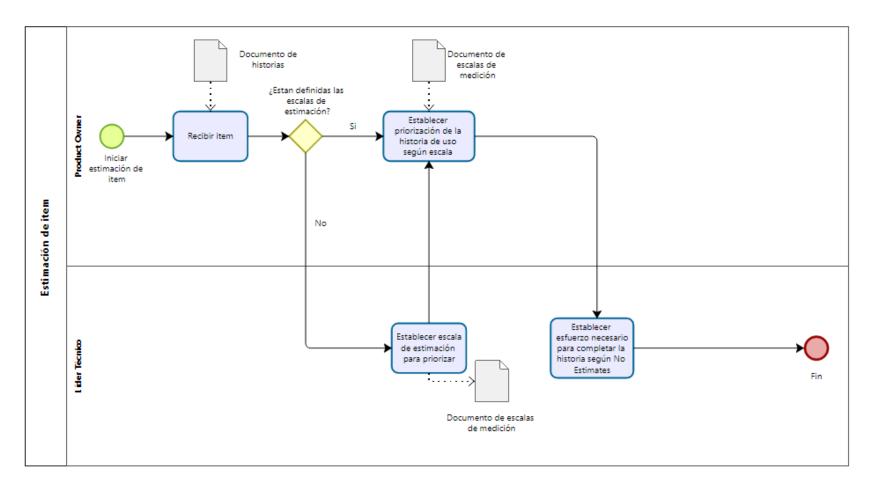


Figura 12. Proceso de estimación de ítems

5.3. Refinamiento del Product Backlog

Descripción

El refinamiento del product backlog es una actividad en la cual el equipo de desarrollo, junto con el Product Owner, revisa y redacta mejoras continuamente de los elementos de la lista de trabajo pendiente (o backlog) del producto. Este proceso ayuda a asegurar que los elementos en el backlog estén bien definidos, priorizados y sean comprensibles para que el equipo pueda abordarlos de manera efectiva en futuras iteraciones o sprints llamando a la gestión del cambio para que revisen las anotaciones y que ellos puedan efectuar el cambio.

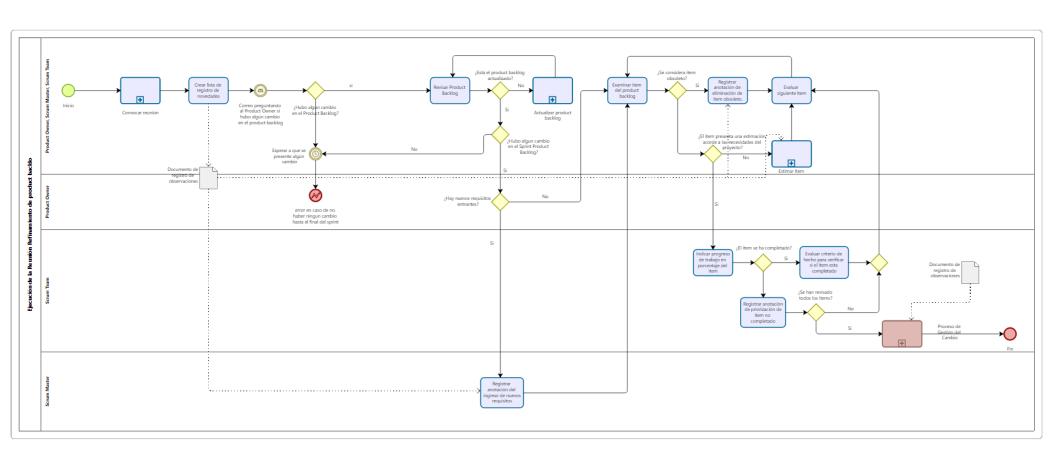


Figura 13. Proceso de Refinamiento del Product Backlog

5.3.1. Elementos del proceso

5.3.1.1. Crear lista de registro de novedades

La Creación del Registro de Novedades para el Proceso de Refinamiento del Product Backlog es una tarea esencial dentro del ciclo de desarrollo ágil, diseñada para asegurar que el equipo de desarrollo y los stakeholders se mantengan informados sobre los cambios, ajustes y novedades que surgen durante el refinamiento del backlog del producto. Esta tarea tiene como objetivo documentar todas las actualizaciones realizadas en el backlog, así como las decisiones tomadas durante las sesiones de refinamiento, lo que permite un seguimiento detallado y transparente de los cambios.

El registro de novedades actúa como un historial de modificaciones que respalda la toma de decisiones futuras y mejora la comunicación dentro del equipo, asegurando que todos tengan una comprensión clara de los cambios y su justificación.

5.3.1.2. Revisar Product Backlog

Este proceso consiste en revisar el Product Backlog para analizar cualquier cambio reciente y determinar si es necesario actualizarlo. El equipo examina cada ítem para confirmar su relevancia, valor de negocio y alineación con las necesidades actuales del proyecto, asegurando que el backlog refleje el estado y las prioridades más recientes del proyecto.

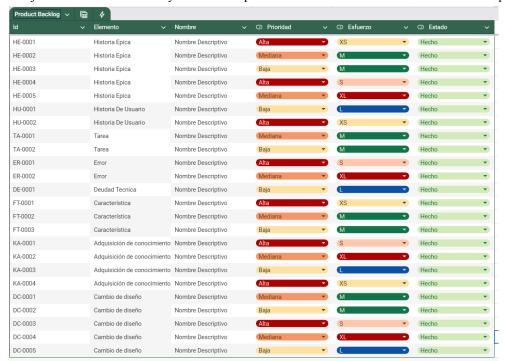


Figura 14. Product Backlog

5.3.1.3. \diamond ¿Está el product backlog actualizado?

Es un condicional que permite preguntar si el product backlog ya se actualizó con anterioridad.

Flujos:

SI

NO

Es un condicional que permite preguntar si hubo cambios en el spring Product Backlog

Flujos:

SI

NO

5.3.1.5. ¿Hay nuevos requisitos entrantes?

Es un condicional que permite preguntar si hay nuevos requisitos entrantes en el product backlog.

Flujos:

SI

NO

5.3.1.6. Registrar anotación del ingreso de nuevos requisitos

Se documentan todos los requisitos nuevos que han sido identificados o solicitados recientemente.



Figura 15. Registro de ingreso de nuevos requisitos al backlog

5.3.1.7. Examinar item del product backlog

Analizar cada ítem para evaluar su valor de negocio y decidir si debe mantenerse, actualizarse, o priorizarse de acuerdo con la estrategia del proyecto.

5.3.1.8. Se considera item obsoleto?

Es un condicional que permite preguntar si el ítem evaluado se considera obsoleto

Flujos:

SI

NO

5.3.1.9. El ítem presenta una estimación acorde a las necesidades del proyecto?

Es un condicional que permite preguntar si el ítem presenta una estimación acorde a las necesidades del proyecto

Flujos

SI

NO

5.3.1.10. Registrar anotación de eliminación de item obsoleto.

Documentar la eliminación de aquellos ítems del Product Backlog que ya no tienen valor de negocio o han sido reemplazados por otros requisitos.



Figura 16. Proceso de anotación por eliminación.

5.3.1.11. Evaluar siguiente ítem

Al concluir el análisis de un ítem, el equipo evalúa el siguiente en la lista, asegurando que la revisión del Product Backlog sea continua y completa.

5.3.1.12. Indicar progreso de trabajo en porcentaje del ítem

Esta tarea implica analizar el estado de cada ítem en el Product Backlog, indicando el porcentaje de avance logrado. Este proceso ayuda a mantener al equipo informado sobre cuánto se ha completado de cada ítem y facilita la gestión del progreso dentro del sprint.

5.3.1.13. Evaluar criterio de hecho para verificar si el item esta completado

El equipo debe revisar si el ítem cumple con la Definición de Hecho (DoD), que es un conjunto de criterios que asegura que el trabajo está realmente terminado y cumple con los estándares requeridos. Este paso confirma que el ítem está completo y listo para ser entregado o integrado.

5.3.1.14. Registrar anotación de priorización de item no completado

Si un ítem no se completa, se debe registrar una anotación que documente la razón y priorizar nuevamente el ítem. Este registro garantiza que el equipo mantenga un seguimiento claro de los elementos pendientes y pueda ajustarlos en futuras iteraciones.

5.3.2. Recursos

- 5.3.2.1. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1
- 5.3.2.2. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2
- 5.3.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3

5.3.2.4. Correo preguntando al Product Owner si hubo algún cambio en el product backlog

Este evento tiene como objetivo asegurar que el equipo de desarrollo y los stakeholders estén al tanto de las actualizaciones recientes en el Product Backlog antes de proceder con las siguientes fases del desarrollo. Este evento es activado durante el proceso de refinamiento y es crucial para mantener la alineación entre el equipo y las prioridades del proyecto, garantizando que se trabajen los elementos más relevantes y actualizados.

5.3.2.5. Esperar a que se presente algún cambio

Permite establecer un período de espera antes de proceder con las siguientes acciones. Este evento se activa justo después de enviar el correo al Product Owner preguntando si ha habido cambios en el backlog. La finalidad de este evento es darle al Product Owner el tiempo necesario para revisar, analizar y responder si se han realizado actualizaciones en el Product Backlog.

Una vez que se ha enviado el correo al Product Owner solicitando confirmación sobre posibles cambios en el Product Backlog, se activa automáticamente el Evento Intermedio de Temporización.

Ahora, si el Product Owner responde antes de que el temporizador expire, el evento se interrumpe automáticamente y el flujo del proceso continúa con la siguiente tarea o evento.

5.3.2.6. Error en caso de no haber ningún cambio hasta el final del sprint

Si al finalizar el sprint no se ha realizado ningún cambio en el Product Backlog, se realiza una evaluación para confirmar que los objetivos del sprint actual siguen siendo relevantes y se mantiene la alineación con las metas del proyecto. En caso de que no se identifiquen cambios y se considere que algunos items podrían necesitar ajustes o aclaraciones, se programará una revisión adicional para el próximo refinamiento del backlog, garantizando la mejora continua.

Error: La falta de cambios en el Product Backlog puede indicar una falta de adaptación a nuevas necesidades o feedback. Se recomienda entonces realizar una revisión retrospectiva con el equipo para identificar posibles mejoras en el proceso de monitoreo y ajuste de requisitos.

Acciones en caso de error:

- Documentar la falta de cambios en el registro de sprint para que el equipo pueda revisarlo en la retrospectiva.
- Revisar la comunicación con stakeholders para confirmar que no haya requisitos pendientes.
- Proponer una reunión adicional con el equipo para asegurar la actualización del Product Backlog antes del próximo sprint, si se considera necesario.

5.3.3. Subprocesos

Este subproceso implica realizar los ajustes necesarios en el Product Backlog para que refleje los cambios en las prioridades y requisitos del proyecto. Durante el refinamiento, el equipo revisa cada ítem para evaluar su valor de negocio y su alineación con los objetivos actuales, determinando si deben añadirse, modificarse o eliminarse elementos según sea necesario. Los cambios en el Product Backlog incluyen la incorporación de nuevos ítems, la actualización de descripciones y criterios de aceptación, la re-priorización de ítems existentes y la eliminación de aquellos que ya no aportan valor.

Pasos del Subproceso:

- 1. Identificar cambios necesarios: Revisar el backlog en busca de items obsoletos o desactualizados y de nuevas necesidades emergentes.
- 2. Modificar ítems existentes: Actualizar descripciones, criterios de aceptación y prioridades de ítems según las necesidades actuales del proyecto.
- 3. Incorporar nuevos ítems: Añadir cualquier requisito o historia de usuario nueva que se haya identificado, documentando detalles y aceptando criterios.
- 4. Eliminar ítems obsoletos: Retirar ítems que ya no tienen valor o no alinean con los objetivos del sprint o del proyecto.
- 5. Re-priorizar el Product Backlog: Asegurar que los ítems de mayor valor estén posicionados para las próximas iteraciones, alineando las prioridades con las metas del proyecto.

Teniendo en cuenta que ya tiene que estar establecido ante la necesidad y el objetivo de la reunión, los temas a tratar y los resultados esperados. obviamente también la fecha y duración tentativas según la disponibilidad general del equipo.

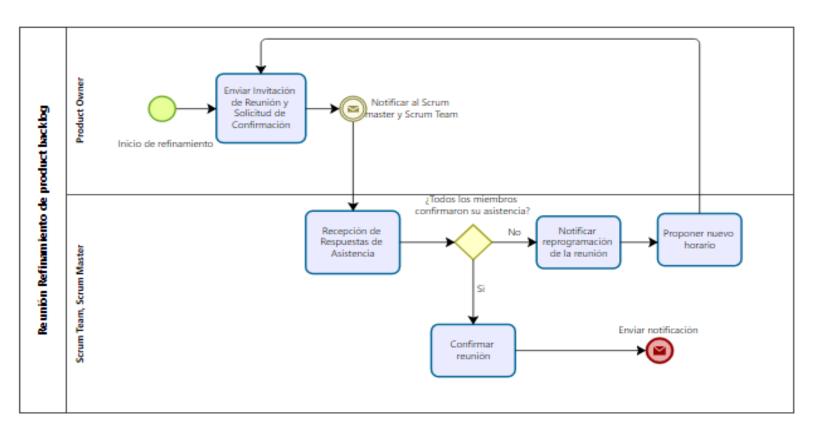


Figura 17. Proceso de convocar reunión

5.3.3.2.1. Enviar Invitación de Reunión y Solicitud de Confirmación

Redactar y enviar una invitación formal a los participantes, detallando el propósito, la fecha y el lugar de la reunión. Incluir una solicitud de confirmación de asistencia para asegurarse de que los miembros puedan asistir.

5.3.3.2.2. Recepción de Respuestas de Asistencia

Monitorear las confirmaciones de asistencia recibidas y llevar un registro para evaluar si es viable proceder con la reunión en la fecha planeada.

5.3.3.2.3. Notificar reprogramación de la reunión

Si no se alcanza el aforo necesario o si hay inconvenientes de disponibilidad, notificar a los participantes sobre la necesidad de reprogramar la reunión.

5.3.3.2.4. Proponer nuevo horario

Sugerir uno o varios horarios alternativos para la reunión y consultar la disponibilidad de los asistentes, optimizando la posibilidad de reunir al equipo completo.

5.3.3.2.5. Confirmar reunión

Una vez se haya definido el horario final con base en las respuestas,se confirma el día y la hora en que se llevará a cabo la reunión.

5.3.3.2.6. Enviar notificación

Enviar una última notificación con la confirmación de la reunión, recordando a los participantes el propósito, lugar, fecha y hora de la misma

5.3.3.3. Proceso de Gestión del Cambio

Aquí convocamos nosotros al proceso de gestión del cambio mandándoles como entrada el documento de registro de novedades para que ellos puedan examinarlo y determinar el hacer los cambios o no.

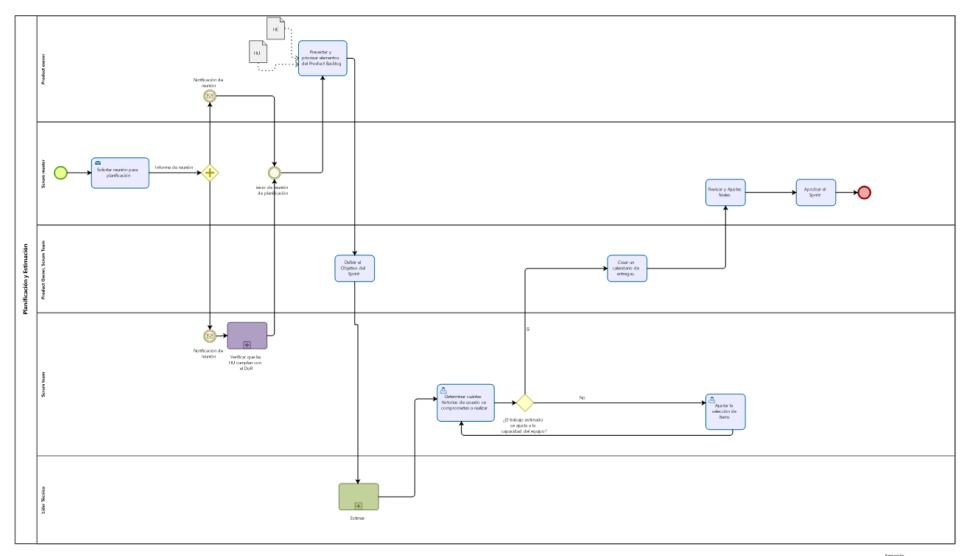




Figura 18. Proceso de planificación y estimación

5.4. Planificación y Estimación

Descripción

La planificación de sprint es el evento Scrum diseñado para asegurarse de que el equipo esté preparado para hacer las cosas correctas en cada sprint. Es la presentación del trabajo que se va a realizar. Este plan es el resultado creado por el trabajo colaborativo de todo el Equipo Scrum.

El product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog. La duración de la mayoría de los eventos de Scrum está relacionada con la duración del sprint. La Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Por lo tanto, para un Sprint más corto debe durar 2 horas por cada semana del Sprint.

La planificación de Sprint aborda los siguientes temas:

- a. ¿Por qué es valioso este Sprint?
- b. ¿Qué se puede hacer en este Sprint?
- c. ¿Cómo se realiza el trabajo elegido?

5.4.1. Elementos del proceso

5.4.1.1. Solicitar reunión para planificación

Descripción

El Scrum Master redacta y envía una invitación formal a los participantes, detallando el propósito, la fecha y el lugar de la reunión. Incluir una solicitud de confirmación de asistencia para asegurarse de que los miembros puedan asistir.

Ejecutantes

Scrum Master

5.4.1.2. Notificación de reunión

Descripción

El Scrum Team recibe la solicitud de la reunión y da respuesta según su disponibilidad.

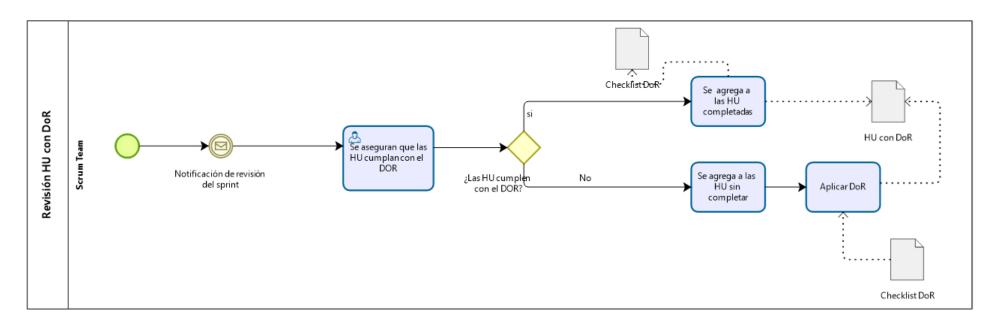




Figura 19. Proceso de revisión HU con DoR.

Descripción

El equipo de desarrollo tiene la obligación de verificar que las historias de usuario que fueron designadas para el sprint actual, cumplan con el criterio de hecho (DoR), si las historias no cumplen con ello, entonces se evalúan nuevamente para los siguientes sprint, según sea necesario.

Ejecutantes

Scrum Team

Responsable

Scrum Team

5.4.1.4. Notificación de reunión

Descripción

El Product Owner recibe la solicitud de la reunión y da respuesta según su disponibilidad

5.4.1.5. Oinicio de reunión de planificación

Descripción

Cuando las partes interesadas han confirmado la asistencia a la reunión y ya asisten a la reunión, por parte del Scrum Master se da inicio a la reunión de planificación y estimación, esta reunión debe durar 2 horas por cada semana del Sprint.

5.4.1.6. Presentar y priorizar elementos del Product Backlog

Descripción

El Product Owner presenta tanto las Historias de Usuario como las Historias Épicas y las socializa con el Scrum Master y el equipo de desarrollo, Asegurándose de que el backlog esté claro y priorizado para que el equipo pueda trabajar.

El Product Owner notifica al Scrum Team sobre los elementos priorizados.

Ejecutantes

Product Owner

Responsable

Product Owner

Consultado

Scrum Team

5.4.1.7. Definir el Objetivo del Sprint

Descripción

El Product Owner y el Scrum Team definen el objetivo del sprint basado en los elementos priorizados presentados por el Product Owner

Ejecutantes

Scrum Team

Responsable

Product Owner

5.4.1.8. **Estimar**

Descripción

La estimación es una puesta en común de los requisitos a lograr, para definir entre todos una suposición lo más exacta posible. El product Owner comparte el documento de historias, a las cuales se le definen una escala de estimación (la escala puede ser de 0... 100, 0... 10, entre otras).

Si se cuenta con líder técnico, este facilita el uso de técnicas como No Estimate, estimando cada historia de usuario del Sprint Backlog y tomando la decisión de calcular la capacidad del equipo para el Sprint.

Existen diferentes técnicas o métodos que se pueden utilizar para hacer la estimación:

a. No Estimate:

Se basa en los siguientes pasos:

- 1. Seleccionar la pieza más importante de trabajo que necesitas terminar (mayor valor primero).
- 2. Dividir el trabajo en partes que no tengan riesgo.
- 3. Desarrollar cada trozo siguiendo la Definition of Done.
- 4. Iterar y refactorizar. En lugar de estimar cada elemento del backlog, se evalúan con una "duración máxima" de la mitad de una iteración.
- b. Scrum Delphi:

Delphi consiste en una técnica que le propone al grupo una estimación anónima de cuánto trabajo implica y cuánto tiempo llevará completar un ítem del Backlog.

Estas estimaciones se revelan y se trazan en un gráfico. Luego, el equipo revela las causas que influyeron en sus estimaciones anónimas, y vuelven a realizar este proceso. Que se repetirá hasta que las estimaciones del equipo estén cerca unas de otras, y se pueda llegar a un consenso para la estimación final.

c. Scrum Póker:

A cada miembro del equipo se le da un juego de tarjetas, con números. Los números suelen estar en la secuencia de Fibonacci: 0, 1, 2, 3, 5, 8, 13 y 21.

El siguiente paso es que el Product Owner revele las historias de usuario del Backlog, es decir los requisitos del cliente. Luego se les pide a los miembros del equipo que definan un nivel de dificultad para ese elemento del Backlog. Para esto, cada miembro debe elegir una carta que muestre el nivel de esfuerzo que creen que tiene la tarea. Los miembros que eligen números altos y bajos, deberán exponer las razones de su elección.

d. Story Points:

Se basa en dar "puntos" a cada historia de usuario, o elemento del backlog, según su nivel de dificultad.

Esta evaluación la realiza el equipo completo, a partir de considerar tres dimensiones del ítem del backlog (o historia de usuario):

- El riesgo.
- La cantidad de esfuerzo requerido.
- El nivel de complejidad.

Una vez que se realiza una evaluación de una historia de usuario, o ítem del backlog, el equipo Scrum puede evaluar otras historias de usuario, en relación con la dificultad que le asignó a esa primera historia.

e. Estimación en Scrum por Afinidad:

La estimación por afinidad es una técnica ideal para estimar rápidamente una gran cantidad de historias de usuario. Utilizando notas adhesivas, fichas o post its, el equipo coloca los ítems del backlog en una pared u otra superficie. Cada miembro del equipo recibe algunas de las fichas que corresponden a cada historia de usuario (ítem del backlog) para colocarlas por tamaño relativo, en un tablero. Esta colocación inicial se realiza en silencio. Una vez que todos han pegado sus Historias de Usuario en la pared, el equipo las revisa y las mueve según las consideraciones generales, y tendrán permitido discutir.

Finalmente, el Product Owner indicará categorías pequeñas, medianas o grandes usando valores de puntos de historia para indicar el tamaño relativo. Como paso final del proceso, se colocará cada tarjeta de forma consensuada en una categoría de tamaño relativo.

f. Técnica de las Camisetas:

Un ejemplo de estimación en Scrum por afinidad es la técnica de las camisetas, donde los miembros del equipo deberán vincular cada ítem analizado, con el tamaño de la camiseta. Si consideran que una actividad es compleja, de riesgo y difícil, la colocarán con el tamaño XL. Por el contrario, una actividad que consideren sencilla, puede colocarse en la categoría de XS.

g. Scrum por cestas o baldes:

Esta técnica que es una derivación de Delphi, consiste en entregar todas las fichas a cada miembro del equipo. Luego se colocan cestas o baldes donde no se pueda ver el interior. Cada cesta tendrá valores relativos a la dificultad: 0,1,2,3,4,5,8,13,20,30,50,100, 200. Las historias, que cada miembro tiene en sus manos, deben colocarse dentro de las cestas que consideren que representa su dificultad. Luego se revela el contenido de cada cesta, si es que las cestas revelan mucha disparidad, se discute a partir de esto, para finalizar con el consenso de cada tarea.

h. Votación por puntos:

Esta técnica sirve para evaluar un conjunto pequeño de historias de usuario, entre 8 y 10. Es muy sencillo, consiste en que cada miembro del equipo recibe una serie de "puntos" a su disposición para que los deje en el ítem que considere más complejo. El ítem que más puntos tenga, será categorizado y eliminado de la lista, para continuar con el proceso.

Ejecutantes

Líder Técnico

5.4.1.9. 🖻 Determinar cuántas historias de usuario se comprometen a realizar

Descripción

El Scrum Team define cuantas historias de usuario se comprometen a realizar en el sprint

Ejecutantes

Scrum Team

Descripción

Es importante conocer la capacidad del equipo, por eso se pregunta si ¿el trabajo estimado se ajusta a la capacidad del equipo?

Flujos

- No
- Si

5.4.1.11. 🚨 Ajustar la selección de ítems

Descripción

Si el trabajo estimado no se ajusta a la capacidad del equipo se hace un ajuste en la selección de ítems o se redefinen las estimaciones.

Una vez hecho esto, se vuelve a definir la cantidad de Historias de Usuario que se comprometen a realizar.

Ejecutantes

Scrum Team

5.4.1.12. Crear un calendario de entregas.

Descripción

Se crea un calendario de entregas para las historias seleccionadas.

Ejecutantes

5.4.1.13. Revisar y Ajustes finales

Descripción

El Scrum Master facilita el cierre de la planificación y verifica que todos los miembros del equipo están alineados con el plan y conocen todos los objetivos.

Ejecutantes

Scrum Master

5.4.1.14. Aprobar el Sprint

Descripción

El Scrum Master confirma el compromiso por parte del equipo y cierra la sesión de planificación, aprobando el Sprint Backlog y el calendario de entregas.

Ejecutantes

Scrum Master

5.4.2. Recursos

- 5.4.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2
- 5.4.2.2. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3
- 5.4.2.3. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1
- 5.4.2.4. Líder Técnico (Rol)

Descripción

Es un miembro con experiencia técnica sólida que, aunque no tiene autoridad formal, asume un rol de guía dentro del equipo en temas relacionados con la arquitectura, diseño y decisiones tecnológicas.

5.5. Implementación

Descripción

Durante la fase de Implementación en Scrum, el Scrum Team trabaja en el desarrollo incremental de las tareas seleccionadas en el Backlog del Sprint. Esto implica llevar a cabo desarrollo de funcionalidades, pruebas, integración de componentes, y gestión de impedimentos. Los avances y bloqueos se revisan en las reuniones diarias (Scrum Diario o Daily Meeting) para asegurar la alineación y adaptabilidad del equipo.

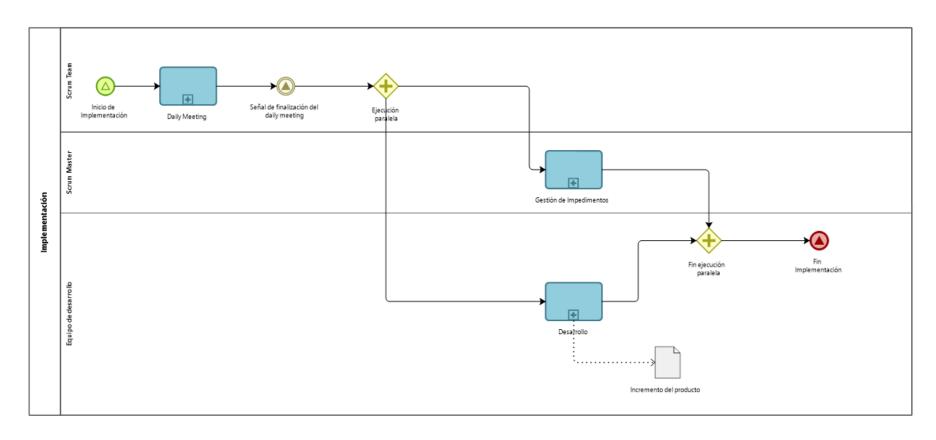


Figura 20. Proceso de implementación.

Ejecutantes

- Scrum Team

Responsable

Scrum Team

5.5.1. Elementos del proceso

5.5.1.1. Daily Meeting

Descripción

La daily meeting, o reunión diaria de Scrum, es una reunión breve que se realiza todos los días con el objetivo de sincronizar al equipo, revisar el progreso diario, identificar posibles bloqueos y planificar el trabajo inmediato. La reunión dura típicamente 15 minutos y sigue una estructura en la cual cada miembro responde tres preguntas clave:

- · ¿Qué hice ayer?
- · ¿Qué haré hoy?
- · ¿Hay algún impedimento en mi camino?

Este proceso contribuye a mantener la transparencia y permite que el equipo ajuste su enfoque de trabajo de manera continua y colaborativa.

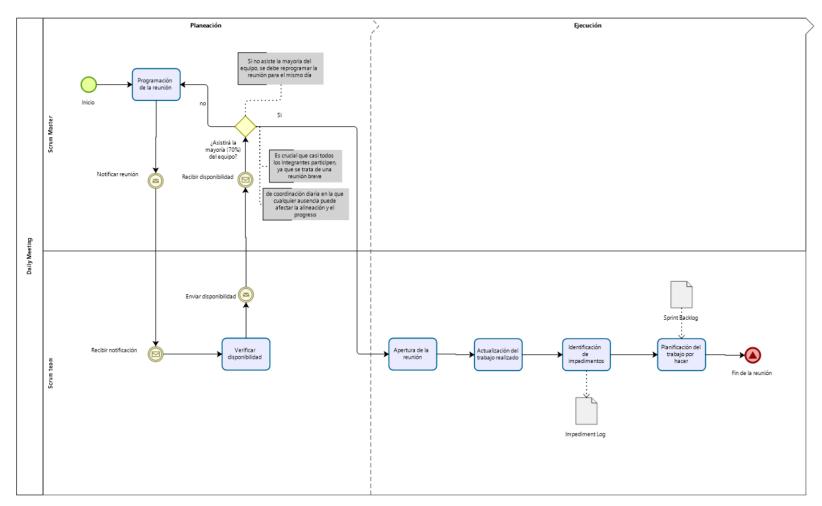




Figura 21. Proceso de Daily Meeting.

Ejecutantes

- Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Master

Informado

- Scrum Team

Frecuencia

- Diaria

Duración

- 15 minutos

5.5.1.1.1. Programación de la reunión

Descripción

El Scrum Master define el horario y lugar de la daily meeting, si es presencial, o el enlace de conexión, si es remota. La daily meeting debe programarse a la misma hora todos los días, para mantener consistencia y facilitar la disponibilidad de los miembros del equipo.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.5.1.1.2. Solutificar reunión

Descripción

El Scrum Master envía un correo a todos los miembros del equipo, notificando la reunión programada.

Datos Requeridos

Lista de asistentes

5.5.1.1.3. Recibir notificación

Descripción

Los participantes reciben el correo de notificación de la reunión.

Descripción

Cada miembro revisa su disponibilidad para la reunión en el horario programado.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

5.5.1.1.5. Enviar disponibilidad

Descripción

Cada miembro del equipo envía un email, confirmando su disponibilidad.

5.5.1.1.6. © Recibir disponibilidad

Descripción

El Scrum Master recibe las respuestas de confirmación de los participantes.

Si la mayoría del equipo confirma asistencia, la reunión procede. Si no se alcanza el quórum, se regresa a la programación de la reunión. No es necesario que estén todos los miembros del equipo, pero sí es importante que estén la mayoría. El ideal es que todo el Scrum Team esté presente para maximizar la coordinación y transparencia diaria. Sin embargo, si no es posible, al menos el 70% del equipo debería asistir. Esto permite que la mayoría de los miembros compartan sus avances, identifiquen bloqueos y sincronicen su trabajo. En la Daily Meeting es crucial que casi todos los integrantes participen, ya que se trata de una reunión breve de coordinación diaria en la que cualquier ausencia puede afectar la alineación y el progreso general del sprint.

Flujos

- Si

5.5.1.1.8. Apertura de la reunión

Descripción

El Scrum Master inicia la daily meeting recordando al equipo los objetivos de la reunión y las reglas, como el enfoque en los tres temas (¿Qué hice ayer?, ¿Qué haré hoy? y ¿Hay algún impedimento?) y la brevedad de la reunión.

Ejecutantes

- Scrum Team

Responsable

Scrum Master

5.5.1.1.9. Actualización del trabajo realizado

Descripción

En esta fase, el Scrum Master va llamando uno a uno a los miembros del equipo, quienes responden a la pregunta "¿Qué hice desde la última reunión?". Cada integrante debe explicar brevemente las tareas completadas y los avances significativos, evitando detalles innecesarios y enfocándose en los puntos importantes que afectan al equipo. Esta actividad permite que todos los miembros tengan visibilidad del progreso del proyecto y de las contribuciones de cada miembro.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Master

Informado

- Scrum Master

Descripción

En esta actividad, el Scrum Master pregunta a cada miembro del equipo si ha enfrentado algún obstáculo que haya dificultado su trabajo o que pueda afectar el cumplimiento de sus tareas futuras. Los participantes deben informar cualquier impedimento, y el Scrum Master los registra para buscar soluciones o asignar responsables de resolución después de la reunión. El objetivo es identificar posibles bloqueos que puedan afectar la productividad del equipo.

Ejecutantes

- Scrum Team

Responsable

Scrum Team

Consultado

- Scrum Master

Informado

Scrum Master

5.5.1.1.11. Planificación del trabajo por hacer

Descripción

En este último paso, el Scrum Master pregunta a cada miembro del equipo "¿Qué vas a hacer hoy?". Los participantes detallan las tareas que planean realizar en el transcurso del día, priorizando actividades y objetivos específicos. Este compromiso diario ayuda a mantener el enfoque en los objetivos inmediatos y contribuye a la organización del flujo de trabajo. Los participantes deben añadir o actualizar sus tareas en el tablero, asegurando transparencia y alineación en el equipo.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Master

Informado

Scrum Master

Descripción

El Scrum Master cierra la reunión, confirmando que todos tienen claridad sobre sus próximas actividades y que los impedimentos han sido documentados para el seguimiento.

5.5.1.1.13. ☐ Planeación

Descripción

La fase de planeación es la etapa inicial del proceso de daily meeting en la que se organiza la logística de la reunión para asegurar que los participantes estén disponibles y se garantice una

asistencia adecuada. Durante esta fase, se programa la reunión, se notifica a los participantes y se confirma su disponibilidad. El objetivo es asegurar que todos los miembros del equipo estén informados sobre la reunión y que puedan asistir, de modo que se logre la máxima efectividad en la siguiente fase.

Descripción

La fase de ejecución es la etapa principal del daily meeting, en la que los miembros del equipo sincronizan su trabajo y planifican el siguiente día de tareas. Durante esta fase, cada miembro comparte sus progresos, se identifican impedimentos y se establecen las prioridades del día. El Scrum Master facilita la reunión para que sea breve y eficiente, y el equipo revisa y actualiza el tablero de tareas según sea necesario.

5.5.1.1.15. Impediment Log

El Impediment Log (Registro de Impedimentos) en Scrum es una herramienta utilizada por el equipo Scrum, especialmente por el Scrum Master, para documentar, rastrear y gestionar los impedimentos que afectan el progreso del equipo. Los impedimentos son cualquier obstáculo o barrera que impide que los miembros del equipo completen su trabajo de manera eficiente durante un sprint. Este registro ayuda a asegurar que todos los problemas que bloquean o ralentizan el trabajo se identifican, se abordan oportunamente y se resuelven, para mantener el flujo de trabajo.

Se propone la siguiente plantilla, que se puede descargar en:



Figura 22. Plantilla del Impediment log.

Descripción

Indica el inicio de 2 subprocesos (Gestión de impedimentos y Desarrollo) de manera paralela.

Descripción

Es una extensión del daily meeting, en la que el Scrum Master trabaja para resolver los bloqueos que afectan el progreso del equipo. Una vez identificados los impedimentos, el Scrum Master los analiza, los prioriza y toma las acciones necesarias para resolverlos o mitigarlos. Este subproceso asegura que los problemas que obstaculizan el flujo de trabajo se resuelvan a tiempo, permitiendo al equipo avanzar de manera efectiva en sus tareas.

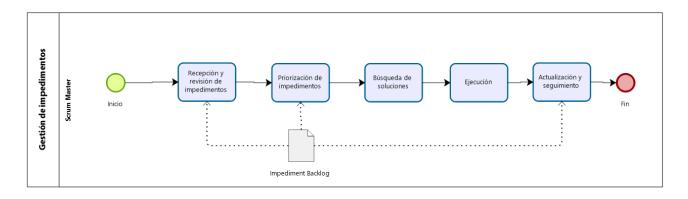




Figura 23. Proceso de gestión de impedimentos.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.5.1.3.1. ■ Recepción y revisión de impedimentos

Descripción

El Scrum Master revisa la lista de impedimentos identificados durante la daily meeting y recoge cualquier información adicional para entender cada problema. Esta revisión inicial permite clasificar los impedimentos según su gravedad y urgencia.

Ejecutantes

Scrum Master

Responsable

- Scrum Master

5.5.1.3.2. Priorización de impedimentos

Descripción

El Scrum Master organiza los impedimentos en orden de prioridad, evaluando su impacto en el proyecto. Los impedimentos que bloquean el progreso de tareas críticas se abordan primero, mientras que los de menor impacto se programan para una resolución posterior.

Ejecutantes

Scrum Master

Responsable

Scrum Master

5.5.1.3.3. Búsqueda de soluciones

Descripción

El Scrum Master analiza posibles soluciones para cada impedimento. Esto puede implicar consultar con otros miembros del equipo, el Product Owner o stakeholders externos, y explorar opciones técnicas o administrativas. En casos complejos, el Scrum Master puede programar reuniones con el equipo o con expertos en el tema.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Product Owner

Informado

- Stakeholders, Product Owner

5.5.1.3.4. □ Ejecución

Descripción

El Scrum Master implementa las soluciones cuando está en su capacidad hacerlo directamente. En caso de que el impedimento requiera recursos o decisiones fuera de su alcance, lo escala al Product Owner o al área correspondiente de la organización.

Ejecutantes

Scrum Master

Responsable

- Scrum Master

Consultado

Product Owner

Informado

- Product Owner

5.5.1.3.5. Actualización y seguimiento

Descripción

El Scrum Master actualiza el tablero de impedimentos, registrando el estado de cada uno y añadiendo notas sobre el progreso o las acciones tomadas. El equipo puede revisar este tablero para conocer el estado de los bloqueos y su resolución.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

5.5.1.3.6. **OFin**

Descripción

Luego de haber solucionado todos los impedimentos y hacer su respectiva gestión

Artefacto descrito en Impediment Log

5.5.1.4. Desarrollo

Descripción

Es donde el equipo de desarrollo convierte los requisitos y objetivos del sprint en código funcional. Aquí, los desarrolladores trabajan activamente en la implementación de las historias de usuario y en la construcción de nuevas funcionalidades, siguiendo las buenas prácticas de programación y asegurando la calidad en cada entrega.

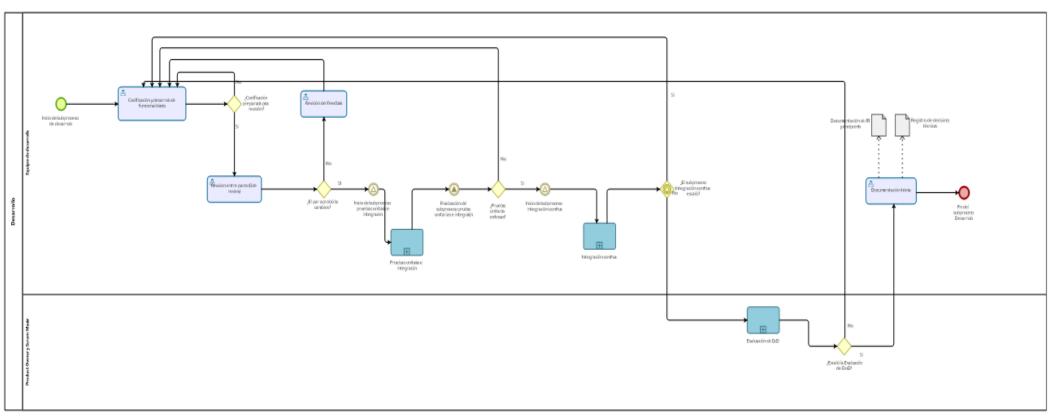




Figura 24. Proceso de desarrollo.

Ejecutantes

- Equipo de desarrollo

Responsable

Equipo de desarrollo

5.5.1.4.1. O Inicio del subproceso de desarrollo

Descripción

Indica el comienzo del trabajo activo de codificación para transformar los requisitos en una solución funcional. En esta fase, el equipo de desarrollo se dedica a implementar las tareas planificadas, basadas en las historias de usuario y los criterios de aceptación definidos previamente. Este subproceso incluye la creación y configuración de módulos, la escritura de funciones y métodos, el desarrollo de interfaces de usuario, y cualquier otra actividad técnica necesaria para construir el producto.

Durante el subproceso de desarrollo, los desarrolladores colaboran constantemente, comparten avances, y realizan ajustes según sea necesario para asegurar que cada incremento esté alineado con los estándares de calidad y la arquitectura establecida.

5.5.1.4.2. 🚨 Codificación y desarrollo de funcionalidades

Descripción

Escribir código limpio y mantenible para implementar las funcionalidades planificadas, asegurando que el código sigue las mejores prácticas de desarrollo, como el uso de patrones de diseño, principios SOLID y técnicas de modularización, para luego, crear y mantener documentación técnica para facilitar futuras modificaciones.

Ejecutantes

- Equipo de desarrollo

Responsable

- Equipo de desarrollo

Informado

- Scrum Master

5.5.1.4.3. ♦¿Codificación preparada para revisión?

Descripción

Se utiliza para verificar si el código desarrollado en un incremento está listo para ser revisado por otros miembros del equipo. Esto implica que el desarrollador considera que el código cumple con los estándares de calidad del equipo, está completo y está listo para su revisión.

Flujos

- No
- Si

5.5.1.4.4. Revision entre pares (Code review)

Descripción

Revisar el código de otros miembros del equipo, evaluando su claridad, eficiencia y adherencia a los estándares del equipo proporcionando retroalimentación constructiva para mejorar el código antes de que se integre al producto. Registrar cambios aprobados y asegurarse de que cumplen con la definición de hecho.

Ejecutantes

- Equipo de desarrollo

Responsable

- Equipo de desarrollo

5.5.1.4.5. **♦** El par aprobó los cambios?

Descripción

Se plantea para confirmar si el código, tras haber pasado por una revisión de pares (code review), ha sido aprobado por otro miembro del equipo. Esto significa que el revisor ha verificado que el código cumple con los estándares de calidad, sigue las buenas prácticas de desarrollo, y respeta los criterios establecidos en la Definición de Hecho (DoD). Además, se asegura de que los cambios realizados no introducen errores ni afectan otras partes del sistema.

Flujos

- No
- SI

5.5.1.4.6. Arevisión del feedback

Descripción

Se realiza cuando el revisor del código (par) ha encontrado problemas o aspectos de mejora en el código que no cumplen con los estándares, y por lo tanto, no aprueba los cambios realizados por el desarrollador. En esta situación, el revisor proporciona retroalimentación detallada, señalando las áreas que deben ajustarse o corregirse, como problemas de lógica, incumplimiento de convenciones de codificación, oportunidades de optimización, o posibles errores que podrían afectar la funcionalidad.

Ejecutantes

- Equipo de desarrollo

Responsable

- Equipo de desarrollo

Informado

Scrum Master

5.5.1.4.7. © Inicio del subproceso pruebas unitarias e integración Descripción

Marca el comienzo de la fase en la que el código desarrollado es sometido a pruebas unitarias para verificar que cada componente o unidad de código funciona correctamente de forma aislada. Este subproceso tiene como objetivo detectar errores en una etapa temprana y asegurar que cada unidad cumple con su funcionalidad esperada y responde adecuadamente a diferentes entradas, sin depender de otras partes del sistema.

5.5.1.4.8. Pruebas unitarias e integración

Proceso

Pruebas unitarias y de integración - Pruebas unitarias e integración

Indica que se ha completado la ejecución de las pruebas unitarias en el código desarrollado y que todas las unidades han pasado satisfactoriamente las validaciones definidas. En este punto, se demuestra que lo desarrollado funciona correctamente de manera independiente y está libre de errores críticos.

5.5.1.4.10. ♦¿Pruebas unitarias exitosas?

Descripción

Se plantea verificar si todas las pruebas unitarias aplicadas al código han sido completadas satisfactoriamente. Este control es fundamental para confirmar que cada unidad del código funciona correctamente y cumple con los requisitos definidos sin errores.

Si la respuesta es afirmativa, significa que el código ha superado todas las validaciones de funcionalidad a nivel unitario y está listo para integrarse con otros componentes o avanzar a fases de revisión y pruebas adicionales. Si la respuesta es negativa, el desarrollador debe corregir los errores encontrados y volver a ejecutar las pruebas hasta obtener resultados exitosos, garantizando que el incremento cumple con los estándares de calidad antes de avanzar.

Flujos

- Si
- No

5.5.1.4.11. © Inicio del subproceso integración continua Descripción

Marca el comienzo de la fase en la cual el código recién desarrollado y probado se integra con el resto del sistema de forma automática y frecuente.

Durante este subproceso, el código se somete a un conjunto de pruebas automatizadas, incluyendo pruebas de integración y de regresión, para verificar que las nuevas implementaciones funcionan bien en conjunto con los componentes existentes. El objetivo es detectar y resolver problemas de compatibilidad de manera temprana, minimizando riesgos y manteniendo una versión actualizada y funcional del software en todo momento.

5.5.1.4.12. Integración continua

Proceso

Integración continua - Integración Continua

Se plantea para verificar si el proceso de integración continua ha detectado problemas durante la combinación del nuevo código con el sistema existente, generando un error o conflicto que impida una integración exitosa. Cuando la integración continua "escala" significa que han surgido problemas serios que requieren la atención inmediata del equipo, tales como conflictos de código, fallos en pruebas automatizadas, o incompatibilidades que podrían afectar la estabilidad del software.

Instanciar

Falso

5.5.1.4.14. **Evaluación DoD**

Proceso

DOD - Evaluación de DoD

5.5.1.4.15. Documentación técnica

Descripción

Se refiere al proceso de registrar información detallada sobre el código y el funcionamiento de los incrementos desarrollados, asegurando que otros miembros del equipo o futuros desarrolladores puedan entender y trabajar con el sistema de manera eficiente. Esta documentación abarca aspectos como la arquitectura del software, detalles de los módulos y componentes, flujos de datos, dependencias, configuraciones y cualquier particularidad técnica relevante para la comprensión y el mantenimiento del código.

Ejecutantes

- Equipo de desarrollo

Responsable

- Equipo de desarrollo

Informado

- Scrum Master

5.5.1.4.16. OFin del subproceso Desarrollo

Descripción

Indica la finalización de la etapa en la cual los desarrolladores han implementado las tareas del sprint, cumpliendo con los requisitos especificados en las historias de usuario y los criterios de

aceptación. Este hito significa que el equipo ha completado la codificación y cualquier otro ajuste necesario en los módulos o funcionalidades asignadas.

5.5.1.4.17. Documentación de API y endpoints Descripción

Contiene información detallada sobre la interfaz de programación de aplicaciones (API) y los endpoints específicos que el sistema ofrece para la comunicación entre aplicaciones o módulos. Este documento describe cada endpoint disponible, especificando:

- URL de cada endpoint: La ruta o dirección que debe usarse para acceder al recurso o servicio.
- Método HTTP: El método de solicitud que el endpoint soporta (como GET, POST, PUT, DELETE).
- Parámetros de entrada: Una lista de parámetros requeridos u opcionales, con sus tipos de datos y descripciones, necesarios para que el endpoint funcione.
- Cuerpo de la solicitud (payload): Estructura y formato de datos que deben enviarse, si corresponde (por ejemplo, en solicitudes POST o PUT).
- Ejemplo de solicitud: Una muestra de cómo debe formarse la solicitud, mostrando el formato JSON o XML esperado.
- Respuesta del servidor: Estructura y ejemplo de la respuesta que el servidor devuelve, con sus posibles valores.
- Códigos de estado HTTP: Los códigos de respuesta que el endpoint puede devolver, indicando si la solicitud fue exitosa o si hubo errores (como 200 OK, 400 Bad Request, 404 Not Found, etc.).
- Descripción de los errores: Detalles sobre los errores comunes y posibles soluciones o mensajes de error específicos.

Se propone una plantilla, la cual se puede descargar en:



Figura 25. Plantilla de documentación de Endpoints.

5.5.1.4.18. Registro de decisiones técnicas

Descripción

Recoge todas las decisiones importantes relacionadas con la tecnología y arquitectura del sistema, proporcionando un historial detallado del proceso de toma de decisiones en el proyecto. Este registro describe:

- Contexto y justificación de cada decisión: Explicación de la situación o problema que motivó la decisión, detallando los desafíos específicos que se buscaban resolver.

- Opciones consideradas: Un listado de las alternativas evaluadas, con sus ventajas, desventajas y limitaciones, proporcionando una visión completa del análisis realizado.
- Decisión final: La opción seleccionada junto con las razones que llevaron al equipo a preferir esa alternativa sobre otras.
- Fecha de la decisión: Indicación del momento en que se tomó la decisión para rastrear el desarrollo temporal del proyecto.
- Implicaciones y consecuencias: Detalles sobre cómo esta decisión impactará el sistema, incluyendo aspectos de desempeño, mantenimiento, costo o futuras modificaciones.
- Responsables: Nombres o roles de las personas que participaron en la toma de la decisión y en su implementación.
- Revisiones o cambios posteriores: Información sobre ajustes o revisiones hechos a la decisión en función de nuevas necesidades o hallazgos durante el desarrollo.

Se propone una plantilla, la cual se puede descargar en:



Figura 26. Plantilla de Registro de decisiones técnicas.

Descripción

Indica el fin de 2 subprocesos (Gestión de impedimentos y Desarrollo) ejecutados de manera paralela

5.5.1.6. Evaluación DoD

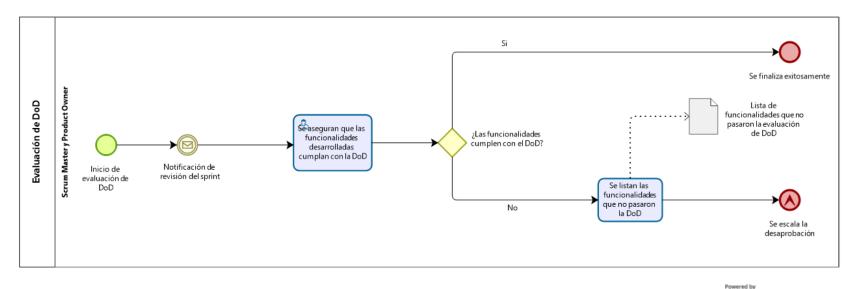




Figura 25. Proceso Evaluación DoD

5.5.1.6.1. Elementos del proceso

5.5.1.6.2. O Inicio de evaluación de DoD

Descripción

Se da inicio al proceso de evaluación de DoD

5.5.1.6.3. Notificación de revisión del Sprint

Descripción

El Scrum Master y el Product Owner reciben la solicitud para realizar la evaluación de las funcionalidades según su disponibilidad

5.5.1.6.4. Se aseguran que las funcionalidades desarrolladas cumplan con la DoD

Descripción

El Scrum Master y el Product Owner se aseguran de que las funcionalidades desarrolladas para el Sprint que se van a socializar, cumplan con el criterio de hecho

Ejecutantes

- Product Owner, Scrum Master

Responsable

Product Owner

5.5.1.6.5. ♦ Las funcionalidades cumplen con el DoD?

Descripción

Para que el equipo pueda garantizar que las nuevas funcionalidades cumplen con el criterio de hecho, es necesario preguntar para cada funcionalidad si se está implementando correctamente lo especificado en la historia de usuario.

Flujos

No

Si

5.5.1.6.6. Se listan las funcionalidades que no pasaron la DoD

Descripción

En esta tarea se agregan todas las funcionalidades que no cumplieron con la DoD en su respectivo artefacto (Lista de funcionalidades que no pasaron la evaluación de DoD)

Ejecutantes

Product Owner, Scrum Master

Responsable

Product Owner

Descripción

Esta finalización indica que alguna de las funcionalidades no cumplió con la DoD

5.5.1.6.8. O Se finaliza exitosamente

Descripción

Esta finalización indica que todas las funcionalidades cumplieron con la DoD

5.5.1.6.9. Lista de funcionalidades que no pasaron la evaluación de DoD

Descripción

Este documento tendrá toda la información de las funcionalidades que no pasaron la evaluación e información adicional como la HU a la cual no le cumplió su propósito.

5.5.1.7. Pruebas unitarias y de integración

Descripción

Constituyen un proceso sistemático que verifica tanto el funcionamiento individual de cada componente como su correcta interacción dentro del sistema, en donde se busca garantizar la calidad del software mediante la validación detallada de funcionalidades específicas y el comportamiento coordinado entre módulos, asegurando que cada parte cumpla sus objetivos de manera aislada y en conjunto con el resto del sistema.

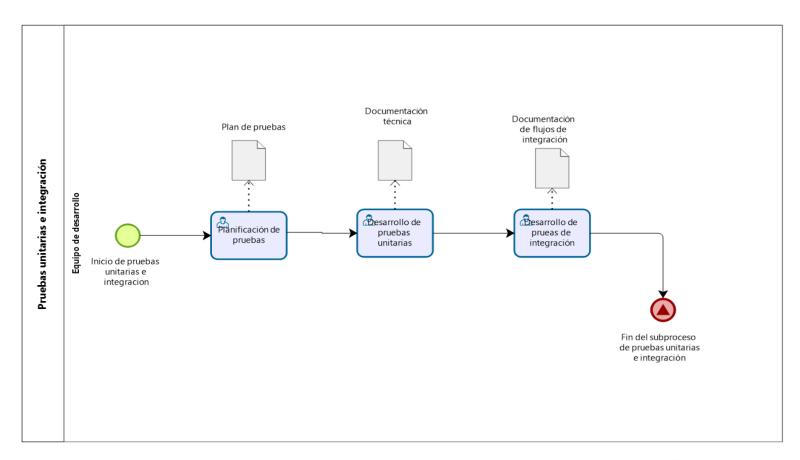




Figura 27. Proceso de Pruebas unitarias e integración.

5.5.1.7.1. Planificación de pruebas

Descripción

Se establece la estructura base del proceso de verificación, definiendo el alcance, recursos necesarios y criterios de aceptación para cada componente, donde durante esta fase se identifican las funcionalidades críticas del sistema, se establecen prioridades y se diseña una estrategia que guiará la ejecución de las pruebas, asegurando una cobertura adecuada y el uso eficiente de los recursos disponibles.

Ejecutantes

- Equipo de desarrollo

Responsable

Equipo de desarrollo

Descripción

El equipo crea y programa pruebas específicas para cada unidad del software y cada componente se somete a pruebas detalladas para confirmar que cumple con su función en aislamiento, en donde se asegura que cualquier error en la lógica de cada módulo se detecte antes de integrar el sistema.

Ejecutantes

- Equipo de desarrollo

Responsable

Equipo de desarrollo

5.5.1.7.3. 🚨 Desarrollo de pruebas de integración

Descripción

Se preparan y configuran las pruebas necesarias para evaluar la interacción entre distintos módulos, se diseñan casos de prueba que verifiquen la comunicación entre componentes, asegurando que el sistema como un todo funcione correctamente y sea estable antes de la fase de integración continua.

Ejecutantes

- Equipo de desarrollo

Responsable

- Equipo de desarrollo

Envío de pruebas desarrolladas, documentación al subproceso de integración continua para su ejecución

5.5.1.7.5. Plan de pruebas

Descripción

Se define el enfoque, los objetivos, el alcance y los recursos necesarios para llevar a cabo las pruebas unitarias e integración

Se propone una plantilla, la cual se puede encontrar aquí: Plantilla plan de pruebas

5.5.1.7.6. Documentación técnica

Descripción

Se incluye todos los documentos, especificaciones y detalles necesarios para definir, ejecutar y comprender las pruebas unitarias desarrolladas

5.5.1.7.7. Documentación de flujos de integración

Descripción

Describe y mapea los flujos de datos y las interacciones entre los módulos para asegurar que se están evaluando correctamente las integraciones del sistema

5.5.1.8. Integración continua

Descripción

La integración continua es una práctica que consiste en integrar y probar automáticamente los cambios en el código de manera frecuente para detectar errores temprano y garantizar su calidad.

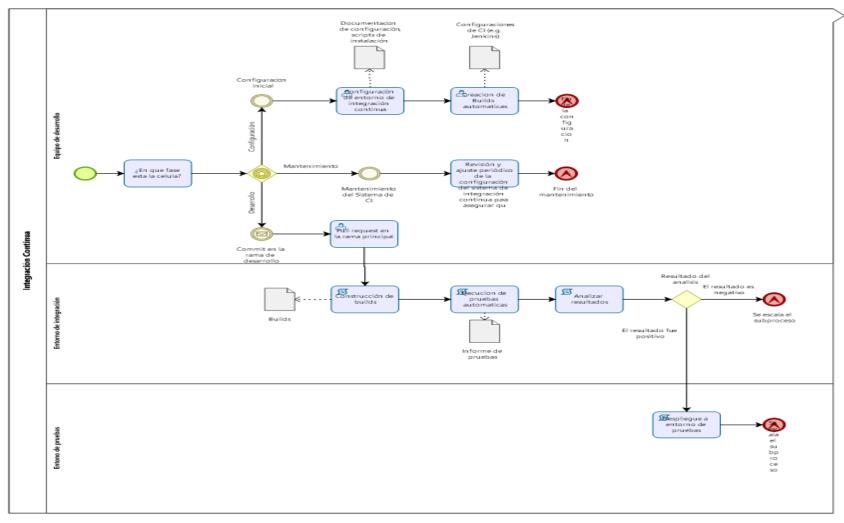




Figura 28. Proceso de integración continua

5.5.1.8.1. En qué fase está la célula?

Descripción

Determina en qué fase del subproceso de integración continua se encuentra la célula SCRUM. Las fases posibles son: configuración del entorno de CI, donde se establecen las herramientas y el pipeline; mantenimiento del entorno de CI, que implica la actualización y solución de problemas; o desarrollo del software, en el que se integran y prueban cambios constantes en el código.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Master

Informado

- Scrum Master

5.5.1.8.2. © Compuerta Basada en Eventos

Descripción

La compuerta paralela que responde a la pregunta "¿En qué fase del subproceso de integración continua se encuentra la célula SCRUM?" se divide en tres caminos: el primero es configuración del entorno de CI, cuando se están instalando y configurando las herramientas de integración continua necesarias; el segundo es mantenimiento del entorno de CI, que abarca la actualización, ajuste y resolución de problemas en el pipeline; y el tercero es desarrollo del software, cuando el entorno ya está operativo y el equipo se enfoca en integrar cambios de código y ejecutar pruebas automáticas.

Instanciar

Falso

5.5.1.8.3. © Commit en la rama de desarrollo

Descripción

El **evento "Commit en la rama de desarrollo"** representa el momento en que los desarrolladores realizan un **commit** (guardar cambios) en la rama de desarrollo del repositorio de código fuente. Este evento desencadena el proceso de integración continua, como la ejecución de pruebas automáticas y la construcción del software. Marca un punto de validación donde los cambios realizados por los desarrolladores son incorporados al flujo de trabajo, asegurando que las nuevas modificaciones sean verificadas y validadas antes de avanzar a etapas posteriores del ciclo de desarrollo.

5.5.1.8.4. Pull request en la rama principal

Descripción

La tarea **Pull request en la rama principal** implica la revisión y fusión de los cambios realizados en una rama de desarrollo hacia la rama principal del repositorio de código. Durante esta tarea, el equipo de desarrollo o los responsables de la revisión de código evalúan las modificaciones propuestas para asegurar que cumplan con los estándares de calidad, funcionalidad y no introduzcan errores. Una vez aprobados, los cambios se integran a la rama principal, lo que permite que el código actualizado pase a las siguientes fases del ciclo de vida del software, como pruebas y despliegue.

Ejecutantes

Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Team

Informado

Scrum Team

5.5.1.8.5. Sonstrucción de builds

Descripción

La tarea **Construcción de builds** consiste en la ejecución automatizada del proceso de construcción del software en el entorno de integración continua. Esta tarea se lleva a cabo cada vez que se realizan cambios en el código, como un **commit** o un **pull request**. Durante la construcción, el sistema de CI compila el código fuente, resuelve dependencias y genera

los artefactos del software, como ejecutables o paquetes. La tarea asegura que el código se construya correctamente en un entorno controlado, validando que no existan errores de compilación y que el sistema esté listo para pasar a las siguientes etapas de pruebas o despliegue.

Ejecutantes

- Entorno de integración

Responsable

- Entorno de integración

Consultado

Repositorio de código

Informado

Scrum Team

Script

Inicio // 1. Consultar el repositorio de código para obtener la última versión ConsultarRepositorio("URL_DEL_REPOSITORIO") // 2. Clonar el repositorio o actualizar la copia local Si (RepositorioNoClonado()) Entonces ClonarRepositorio("URL_DEL_REPOSITORIO") Sino ActualizarRepositorio() FinSi // 3. Instalar dependencias necesarias InstalarDependencias() // 4. Compilar el código fuente CompilarCodigo() // 5. Generar artefactos de la build GenerarBuild() // 6. Subir artefactos al repositorio de builds SubirArtefactos("RUTA_DEL_REPOSITORIO_DE_BUILDS") Imprimir("La build fue construida con éxito y los artefactos fueron almacenados.")Fin

5.5.1.8.6.

Ejecución de pruebas automáticas

Descripción

La tarea **Ejecución de pruebas automáticas** consiste en ejecutar un conjunto de pruebas previamente definidas en el entorno de integración continua para verificar la calidad del código. Estas pruebas se ejecutan de manera automatizada cada vez que se realiza un **commit** o una **construcción de build**, y pueden incluir pruebas unitarias, de integración, de rendimiento, entre otras. El objetivo es identificar errores o fallos en el código de forma temprana, asegurando que los cambios no afecten negativamente la funcionalidad del sistema y cumplan con los estándares de calidad establecidos.

Ejecutantes

Entorno de integración

Responsable

- Entorno de integración

Consultado

- Repositorio de código

Informado

Scrum Team

Script

Inicio // 1. Obtener la build generada ObtenerBuild("RUTA_DE_LA_BUILD") // 2. Ejecutar pruebas automáticas sobre EjecutarPruebasAutomatizadas("RUTA DE LA BUILD") // 3. Verificar resultados de las pruebas Si (TodasLasPruebasExitosas()) Entonces // 4. Generar informe de pruebas exitosas GenerarInformePruebas("Resultado: **Todas** las pruebas pasaron", "RUTA INFORME PRUEBAS") Sino // 5. Generar informe de pruebas fallidas GenerarInformePruebas("Resultado: Algunas pruebas fallaron". "RUTA INFORME PRUEBAS") FinSi // 6. Subir informe de pruebas al repositorio de artefactos SubirArtefacto("RUTA INFORME PRUEBAS") Imprimir("Las pruebas fueron ejecutadas y el informe de pruebas fue generado.")Fin

5.5.1.8.7.



Analizar resultados

Descripción

La tarea **Analizar resultados** se ejecuta de manera automatizada en el entorno de integración continua después de que se han ejecutado las pruebas automatizadas. Su objetivo es evaluar los resultados obtenidos de las pruebas (por ejemplo, pruebas unitarias, de integración o de rendimiento) y determinar si el código cumple con los criterios de calidad establecidos. Esta tarea incluye la recopilación y el análisis de los logs de las pruebas, la comparación de los resultados obtenidos con los resultados esperados, y la identificación de cualquier error o fallo. El análisis de los resultados permite generar reportes detallados que ayudan al equipo de desarrollo a identificar áreas de mejora y garantizar que el software esté listo para las siguientes etapas del ciclo de vida del desarrollo.

Ejecutantes

- Entorno de integración

Responsable

- Entorno de integración

Consultado

- Repositorio de código

Informado

Scrum Team

Script

// 1. Obtener Inicio resultados los de las pruebas ejecutadas ObtenerResultadosDePruebas("RUTA RESULTADOS PRUEBAS") // 2. Analizar los resultados de las pruebas Si (ResultadosSonExitosos()) Entonces // 3. Informar de éxito Imprimir("Las pruebas fueron exitosas.") Sino // 4. Informar de fracaso Imprimir("Algunas pruebas han fallado.") FinSiFin

5.5.1.8.8.



Resultado del análisis

Descripción

La **compuerta de análisis de resultados** evalúa los resultados de las pruebas automatizadas y determina el siguiente paso en el proceso. Si los resultados son satisfactorios, el flujo continúa con el despliegue de la build en un entorno de pruebas para su validación más detallada. Sin embargo, si los resultados son insatisfactorios, el proceso se escala, lo que implica que el flujo se redirige a un nivel superior para una revisión más profunda y la corrección de los errores antes de intentar nuevamente las pruebas.

Flujos

El resultado es negativo

El resultado fue positivo

5.5.1.8.9.



Despliegue a entorno de pruebas

Descripción

La tarea **Despliegue a entorno de pruebas** consiste en la implementación automatizada de la build generada en un entorno de pruebas controlado. Este proceso toma el artefacto construido y lo despliega en un entorno que simula las condiciones de producción para validar su funcionalidad en un contexto más realista. El despliegue incluye la configuración de bases de datos, servidores y otras dependencias necesarias para ejecutar la aplicación, permitiendo realizar pruebas de integración y validación antes de que el software se despliegue en producción. Esta tarea se ejecuta de manera automática, asegurando rapidez y consistencia en cada ciclo de integración continua.

Ejecutantes

Entorno de integración

Responsable

Entorno de pruebas

Consultado

Repositorio de código

Informado

Scrum Team

Script

Inicio // 1. Obtener la build generada ObtenerBuild("RUTA DE LA BUILD") // 2. Configurar el entorno de pruebas ConfigurarEntornoDePruebas() // 3. Desplegar la build DesplegarEnEntornoDePruebas("RUTA DE LA BUILD", en el entorno de pruebas "RUTA ENTORNO DE PRUEBAS") // 4. Verificar si el despliegue fue exitoso Si (DespliegueExitoso()) Entonces Imprimir("Despliegue a entorno de pruebas completado con éxito.") Sino Imprimir("Error en el despliegue a entorno de pruebas.") FinSiFin

El resultado fue positivo

5.5.1.8.10.



🕙 Se escala el subproceso

Descripción

El evento de finalización escala el subproceso desde el entorno de pruebas, indica que el proceso de despliegue en el entorno de pruebas ha concluido correctamente, sin necesidad de intervenciones adicionales. Este evento marca el fin del subproceso de despliegue, y el flujo se escala hacia un nivel superior para continuar con otras tareas o fases del ciclo de vida del desarrollo. El escalamiento asegura que el proceso de integración continua siga avanzando, delegando la responsabilidad a otros equipos o etapas, si es necesario, para gestionar lo que sigue en el proceso.

El resultado fue positivo

5.5.1.8.11.



Se escala el subproceso

Descripción

El evento "Se escala el subproceso" en el contexto de la fase de integración continua indica que el proceso de integración no ha podido completarse con éxito debido a errores significativos o fallos que no pueden ser resueltos dentro del flujo estándar. Este evento marca el fin del proceso de integración en su nivel actual y activa un mecanismo de escalamiento, donde el flujo se redirige a un nivel superior para realizar una revisión más detallada o para que un equipo especializado solucione los problemas encontrados. El objetivo es asegurar que el proceso de integración continúe de manera efectiva tras la intervención necesaria.

5.5.1.8.12.



Configuración inicial

Descripción

El evento de configuración inicial marca el inicio del subproceso de integración continua y se activa cuando la célula SCRUM comienza la configuración del entorno de CI. En este evento, se establecen las herramientas necesarias para el pipeline, como servidores, repositorios y sistemas de automatización. Este paso es crucial para preparar el entorno donde se integrarán los cambios de código y se ejecutarán las pruebas continuas, asegurando que todo esté listo para las fases posteriores del proceso de integración continua.

El resultado fue positivo

5.5.1.8.13.



Configuración de entorno de integración continua

Descripción

Se deben documentar los pasos de configuración y cualquier variable de entorno necesaria. El entorno debe ser lo más similar posible al de producción para evitar problemas de integración.

Ejecutantes

Scrum Team

Responsable

Scrum Team

Consultado

Scrum Master

Informado

Scrum Master

El resultado fue positivo

5.5.1.8.14.



Creación de Builds automáticas

Descripción

Utilizar herramientas de CI/CD para automatizar la creación de builds. Se debe asegurar que las builds incluyan todos los pasos necesarios para compilar el proyecto correctamente.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

Scrum Master

Informado

Scrum Master

El resultado fue positivo

5.5.1.8.15.



Fin de la configuración

Descripción

El evento de finalización de la fase de configuración marca el término de la fase inicial en el proceso de integración continua. En este punto, todas las herramientas, entornos y configuraciones necesarias han sido establecidas y están operativas, lo que permite que el equipo pueda comenzar con las siguientes fases del ciclo de integración continua, como el desarrollo y la integración del código. Este evento indica que la configuración ha sido completada con éxito y el entorno está listo para ser utilizado en los procesos de construcción, pruebas y despliegue continuos.

El resultado fue positivo

5.5.1.8.16.



Mantenimiento del Sistema de CI

Descripción

El **evento de Mantenimiento del Sistema de CI** representa el proceso continuo de actualización, ajuste y optimización del entorno de integración continua. Este evento abarca actividades como la actualización de herramientas, la solución de problemas en el pipeline, la mejora de la infraestructura y la revisión de configuraciones para garantizar que el sistema de CI funcione de manera eficiente y sin interrupciones. Se asegura de que el entorno de CI siga siendo robusto, adaptable a nuevas necesidades y capaz de soportar la evolución del desarrollo del software.

El resultado fue positivo

5.5.1.8.17. Revisión y ajuste periódico de la configuración del sistema de integración continua

Descripción

Debe haber un procedimiento establecido para revisar y actualizar el entorno y las herramientas de CI, asegurando que se adapten a las necesidades del equipo.

Ejecutantes

Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Master

Informado

Scrum Master

El resultado fue positivo

Descripción

El **evento de fin del mantenimiento** marca la conclusión de las actividades de mantenimiento del sistema de integración continua. Este evento indica que todas las tareas de actualización, optimización y solución de problemas han sido completadas con éxito, y el sistema de CI está nuevamente en funcionamiento de manera eficiente. Al finalizar este evento, el entorno está listo para continuar con las operaciones normales, garantizando que el proceso de integración continua siga siendo estable y fiable.

El resultado fue positivo

5.5.1.8.19. Documentación de configuración, scripts de instalación

Descripción

El artefacto **Documentación de configuración, scripts de instalación** contiene toda la información necesaria para reproducir y configurar el entorno de integración continua.

Incluye instrucciones detalladas sobre la instalación de herramientas, configuraciones específicas del entorno, y scripts automatizados que permiten la instalación y configuración rápida del pipeline de CI. Este artefacto asegura que el proceso de configuración sea repetible y consistente en diferentes entornos o para nuevos miembros del equipo.

El resultado fue positivo

5.5.1.8.20. Configuraciones de CI (e.g. Jenkins)

Descripción

El artefacto **Configuraciones de CI (e.g. Jenkins)** incluye los archivos y parámetros necesarios para la configuración y personalización del sistema de integración continua, como Jenkins u otra herramienta similar. Este artefacto contiene los pipelines definidos, trabajos automatizados, reglas de compilación, pruebas, despliegue, y cualquier otro ajuste relevante para garantizar que el entorno de CI funcione correctamente. Facilita la correcta operación y mantenimiento del sistema de CI, permitiendo la ejecución automatizada y eficiente del proceso de integración continua.

El resultado fue positivo

5.5.1.8.21. Builds

Descripción

El **artefacto "Build"** es el resultado del proceso de construcción de un software a partir de su código fuente. Este artefacto es una versión compilada y lista para ser probada, desplegada o distribuida. La build incluye todos los archivos necesarios para ejecutar la aplicación, como binarios, bibliotecas, configuraciones y otros componentes que componen la aplicación final. Dependiendo del ciclo de vida del desarrollo, las builds pueden pasar por diferentes fases, como pruebas automatizadas, revisión de calidad y despliegue en entornos específicos. Este artefacto es crucial para asegurar que el software está en un estado funcional y listo para su siguiente paso en el proceso de desarrollo.

El resultado fue positivo

5.5.1.8.22. Informe de pruebas

Descripción

Muestra que pruebas fallaron, cuales pasaron, muestra logs de la ejecución de las pruebas para facilitar la solución de errores

El resultado fue positivo

5.5.1.8.23. Incremento del Producto

Se refiere al conjunto de todos los elementos completados durante un sprint, más el trabajo de incrementos previos, que están completamente finalizados y listos para ser entregados al cliente o usuarios finales.

5.5.2. Recursos

5.5.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2

5.5.2.2. Equipo de desarrollo (Rol)

El equipo de desarrollo en Scrum es un grupo de profesionales encargados de construir y entregar incrementos del producto con calidad en cada sprint. Este equipo se caracteriza por ser autoorganizado e interdisciplinario, lo que significa que decide cómo realizar su trabajo y cuenta con todas las habilidades necesarias para completar las tareas sin depender de recursos externos. Su objetivo principal es crear un producto funcional y valioso, cumpliendo con los estándares definidos. Además, colabora estrechamente con el Product Owner para entender y priorizar los requisitos, y con el Scrum Master para superar obstáculos. Trabajan de manera conjunta durante todo el ciclo del sprint, manteniéndose enfocados en el objetivo definido y participando activamente en todas las ceremonias de Scrum.

- 5.5.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3
- 5.5.2.4. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1
- 5.5.2.5. Stakeholders (Rol) Ya especificado en el inicio 4.8.2.4

5.6. Revisión del Sprint

Descripción

La revisión de sprint es el evento Scrum donde todo el trabajo completado durante el sprint. El propósito de la Revisión del Sprint es inspeccionar el resultado del Sprint y determinar las adaptaciones futuras. El Equipo Scrum presenta los resultados de su trabajo a las partes interesadas clave y el progreso hacia el Objetivo del Producto es discutido.

Dura 1 hora por semana del sprint. Lo que quiere decir, si un sprint dura 2 semanas, la reunión de la revisión debería durar 2 horas. Durante el evento, el Equipo Scrum y las partes interesadas revisan lo que se ha logrado en el Sprint y lo que ha cambiado en su entorno. A partir de esta información, los asistentes colaboran sobre qué hacer a continuación. El

Product Backlog también puede ajustarse para satisfacer nuevas oportunidades.

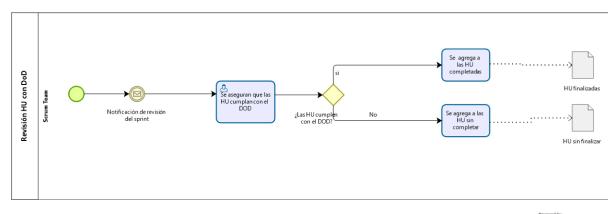




Figura 29. Proceso de Revisión HU con DoD

5.6.1. Elementos del proceso

5.6.1.1. Revisión HU con DoD

Descripción

Este proceso lo hace el equipo de desarrollo o Scrum Team, antes de la reunión de revisión del Sprint para verificar cuáles Historias de Usuario cumplen con el criterio de hecho (DoD), lo que quiere decir que verifican qué Historias de Usuario están completamente finalizadas y cuáles se deben evaluar nuevamente para futuros Sprints.

5.6.1.1.1. Notificación de revisión del sprint

Descripción

El Scrum Team recibe la solicitud de la reunión y da respuesta según su disponibilidad

5.6.1.1.2. Se aseguran que las HU cumplan con el DOD Descripción

El Scrum Team se asegura de que las Historias de Usuario que están para el Sprint que se va a socializar, cumplan con el criterio de hecho

5.6.1.1.3. \bigcirc ¿Las HU cumplen con el DOD?

Descripción

Para que el equipo pueda garantizar que las Historias de Usuario cumplen con el criterio de hecho, es necesario preguntar si cada una de las historias del Sprint.

Flujos

- Si
- No

5.6.1.1.4. Se agrega a las HU completadas

Descripción

Si la respuesta es no, entonces se agrega al artefacto que contiene las historias de usuario sin finalizar para que se evalúan nuevamente para el siguiente sprint

5.6.1.1.5. □ Se agrega a las HU sin completar

Descripción

Son las historias de usuario que no están del todo terminadas, para ser reevaluadas en el siguiente sprint.

5.6.1.1.6. HU sin finalizar

Descripción

Son las historias de usuario que no están del todo terminadas, para ser reevaluadas en el siguiente sprint.

5.6.1.1.7. HU finalizadas

Descripción

Son las historias de usuario que se realizaron por completo durante el sprint

5.6.1.2. Revisión del Sprint

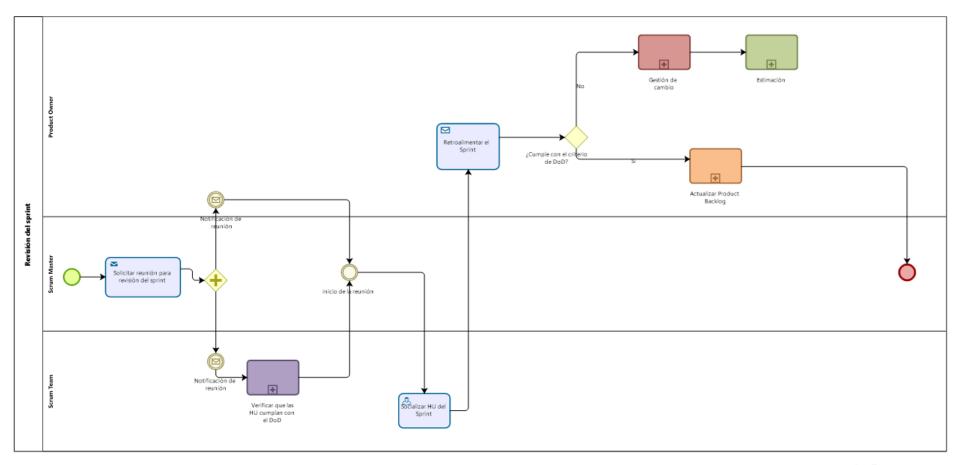




Figura 30. Proceso de Revisión del sprint.

5.6.1.2.1. Solicitar reunión para revisión del sprint

Descripción

El Scrum Master envía la solicitud de la reunión para planificación y estimación al Scrum Team y Product Owner

Ejecutantes

Scrum Master

Implementación

Servicio Web

5.6.1.2.2. Notificación de reunión

Descripción

El Scrum Team recibe la solicitud de la reunión y da respuesta según su disponibilidad.

5.6.1.2.3. Garantizar que las HU cumplan con el DoD

Descripción

El Scrum Team se asegura de que las Historias de Usuario que estan para el Sprint que se va a socializar, cumplan con el criterio de hecho

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Product Owner

5.6.1.2.4. Notificación de reunión

Descripción

El Product Owner recibe la solicitud de la reunión y da respuesta según su disponibilidad.

5.6.1.2.5. Socializar HU del Sprint

Descripción

El Scrum Team socializa lo que hizo durante el sprint

Ejecutantes

Scrum Team

5.6.1.2.6. ■ Retroalimentar el Sprint

Descripción

Basado en la presentación hecha por el scrum Team, el product owner hace una retroalimentación para posibles mejoras y se examina si todas las historias de usuario tienen el criterio de hecho.

Implementación

Servicio Web

5.6.1.2.7. ♦ Cumple con el criterio de DoD?

Descripción

Se plantea la pregunta

Flujos

- Si
- No

5.6.1.2.8. Se actualiza el sprint

Descripción

Si cumple, entonces se califica la historia como completada y se elimina del sprint y se actualiza el sprint

5.6.1.2.9. Product Backlog Repriorizado

Descripción

Se hace la actualización del product backlog para futuros sprints

5.6.1.2.10. Gestión de cambio

Descripción

Para las HU que no cumplan, se devuelven al product backlog para ser reprocesadas en futuros sprints

5.6.1.2.11. **Estimación**

Descripción

El líder técnico facilita el uso de técnicas como No Estimate, estimando cada historia de usuario del Sprint Backlog y toma la decisión de calcular la capacidad del equipo para el Sprint

5.6.2. Recursos

- 5.6.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2
- 5.6.2.2. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3
- 5.6.2.3. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1

5.7. Lanzamiento (Demo)

Descripción

Es una reunión clave donde el equipo presenta el incremento de producto desarrollado durante el sprint. Su objetivo principal es mostrar el avance y los resultados obtenidos, brindando la oportunidad para que los stakeholders (partes interesadas) proporcionen retroalimentación. Durante esta reunión, el Scrum Team presenta las funcionalidades completadas, discute las dificultades encontradas y documenta las ideas que surgen.

Este evento también es un momento crucial para la actualización del Product Backlog, ya que el Product Owner, con base en los comentarios recibidos, puede realizar ajustes de prioridad o agregar nuevos elementos al backlog. Además, el Product Owner ofrece una revisión general del trabajo realizado y discute con el equipo cómo el incremento aporta al objetivo general del producto.

Aunque el evento de Demo tiene una duración recomendada de 1 hora, su tiempo exacto puede variar dependiendo de la complejidad del incremento y la cantidad de retroalimentación. Lo esencial es que todos los involucrados tengan claridad sobre el avance y los pasos siguientes en el proceso de desarrollo.

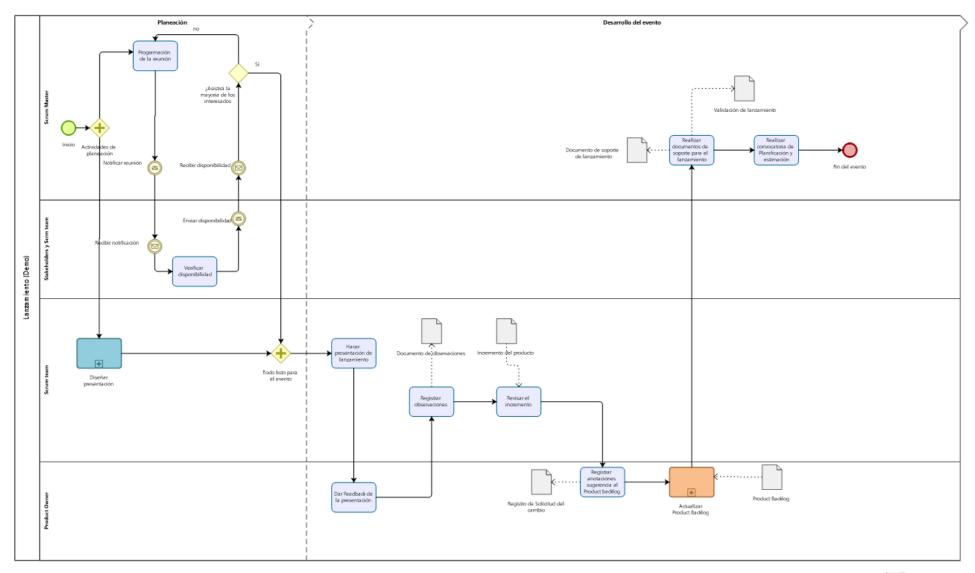




Figura 31. Proceso de Lanzamiento (Demo).

Ejecutantes

- Scrum Team

Responsable

Scrum Master

Consultado

Product Owner

Informado

Stakeholders

Duración

- 1 hora por semana de duración del sprint

5.7.1. Elementos del proceso

5.7.1.1. Lanzamiento

5.7.1.1.1. **OInicio**

Descripción

Este es el evento inicial donde comienza todo el proceso del lanzamiento (demo) del sprint. Marca el comienzo de la planificación y las actividades que aseguran la correcta ejecución de la Demo.

Descripción

Este paso genera dos flujos paralelos que se ejecutan simultáneamente. Por un lado, el Scrum Master se encarga de organizar la reunión, asegurando que todos los participantes clave estén disponibles, mientras que, por otro lado, el Scrum Team y el Product Owner diseñan y preparan la presentación que se mostrará durante la reunión.

5.7.1.1.3. Programación de la reunión

Descripción

En esta actividad, el Scrum Master es responsable de organizar la Demo, lo que incluye determinar la fecha, la hora y los participantes. La programación se basa en la disponibilidad de los miembros del equipo y los stakeholders, asegurando que todos los interesados clave estén disponibles para asistir a la reunión.

Ejecutantes

- Scrum Master

Responsable

Scrum Master

Consultado

Product Owner

Informado

Scrum Team, Stakeholders

5.7.1.1.4. Notificar reunión

Descripción

Una vez que la reunión está programada, el Scrum Master envía un correo electrónico a todos los participantes confirmados, incluyendo a los miembros del Scrum Team y Stakeholders clave. Este correo debe contener todos los detalles esenciales de la reunión: fecha, hora, lugar (o enlace si es virtual), así como una breve agenda de lo que se discutirá. Esta notificación tiene como objetivo garantizar que todos los interesados tengan suficiente tiempo para prepararse y asistir a la reunión.

5.7.1.1.5. Recibir notificación

Descripción

Este evento ocurre cuando los participantes reciben la notificación enviada por el Scrum Master. El correo contiene toda la información relevante para asistir a la reunión, lo que les permite confirmar su disponibilidad para participar en la Demo.

Descripción

Después de recibir la notificación, cada uno de los participantes (tanto Scrum Team como Stakeholders) verifica su disponibilidad para asistir a la Demo. Esta es una actividad activa donde cada miembro debe revisar su calendario o situación personal para confirmar si puede asistir a la reunión. Si un participante no puede asistir, deberá informar al Scrum Master lo antes posible.

Eiecutantes

- Stakeholders, Scrum Team

Consultado

- Scrum Master

5.7.1.1.7. Enviar disponibilidad

Descripción

Una vez que los participantes han verificado su disponibilidad, cada uno de ellos envía al Scrum Master una confirmación, indicando si pueden asistir o no. Este evento garantiza que el Scrum Master tenga la información necesaria para evaluar si se puede proceder con la reunión.

5.7.1.1.8. Recibir disponibilidad

Descripción

El Scrum Master recibe las confirmaciones de todos los interesados y miembros del Scrum team para tomar la decisión de si se lleva a cabo o no en la fecha y hora programadas.

El Scrum Master evalúa si la mayoría de los participantes están disponibles para asistir a la Demo. Si la mayoría está disponible (Se recomienda que al menos se encuentre el 80% del Scrum Team, dentro del cual, es indispensable la asistencia del Product Owner y del Scrum Master), se procede con la reunión; si no, el flujo vuelve a la actividad de programación para reorganizar la reunión en un horario diferente.

Flujos

- no
- Si

5.7.1.1.10. Diseñar presentación

Descripción

El Scrum Team prepara la presentación que se utilizará en la Demo. El equipo debe decidir qué funcionalidades se presentarán y cómo se organizará la información. La presentación debe centrarse en mostrar el incremento del producto desarrollado durante el sprint, destacando los logros y los puntos más importantes del trabajo completado. Además, el equipo debe anticipar las preguntas que los stakeholders puedan tener y preparar respuestas o puntos adicionales que ayuden a clarificar el progreso del proyecto.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Product Owner

Informado

Stakeholders

Proceso

- Diseñar Presentación - Diseñar Presentación

5.7.1.1.11. Todo listo para el evento

Descripción

Esta compuerta lógica indica que ambas ramas paralelas, es decir, la programación de la reunión y el diseño de la presentación, han terminado. Esto marca el punto de transición en el proceso donde el equipo está listo para avanzar al siguiente paso, que es la realización del evento de lanzamiento. Al haber completado la programación de la reunión y diseñado la presentación, el equipo ya está preparado para poner en marcha la Demo y continuar con el flujo de trabajo.

5.7.1.1.12. Hacer presentación de lanzamiento

Descripción

Una vez que la presentación ha sido diseñada, el equipo (principalmente el Scrum Master y los miembros clave del Scrum Team) procede a realizar la presentación de la Demo. Esto incluye prepararse para la presentación en sí, asegurándose de que todos los materiales estén listos (como la presentación, las demostraciones del software y cualquier documentación que se vaya a compartir). La presentación debe ser clara, concisa y resaltar los logros del sprint, así como los incrementos realizados.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Product Owner

Informado

- Stakeholders, Scrum Team

5.7.1.1.13. Dar Feedback de la presentación

Descripción

Después de la presentación, el Product Owner proporciona retroalimentación al Scrum Team. Este feedback debe ser constructivo, destacando lo que se ha hecho bien y las áreas de mejora. Es crucial que el Product Owner sea claro sobre si los puntos cubiertos en la Demo corresponden con las expectativas del cliente o los stakeholders y si el trabajo realizado está alineado con las necesidades del negocio.

Ejecutantes

Product Owner

Responsable

- Product Owner

Consultado

- Scrum Master

Informado

- Scrum Team

5.7.1.1.14. Registrar observaciones

Descripción

Tras recibir el feedback del Product Owner, el Scrum Team registra las observaciones y comentarios dados. Esta actividad es importante porque asegura que las sugerencias y críticas sean documentadas para que el equipo pueda revisarlas más tarde y utilizarlas en el próximo ciclo de trabajo. Las observaciones registradas deben ser revisadas para asegurarse de que se comprendan correctamente y que se puedan implementar en el futuro.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Product Owner

Informado

- Scrum Master

5.7.1.1.15. Revisar el incremento

Descripción

Después de registrar las observaciones, el Scrum Team revisa el incremento del producto presentado en la Demo. Esto implica asegurarse de que lo mostrado durante la reunión cumple con los criterios de aceptación establecidos al inicio del sprint. El equipo también debe revisar si el incremento realmente aporta valor al cliente y si se puede considerar como un avance significativo en el desarrollo del producto.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Product Owner

Informado

- Scrum Master

5.7.1.1.16. Registrar anotaciones sugerencia al Product backlog

Descripción

La revisión del incremento del producto permite capturar sugerencias, ideas o mejoras directamente de quienes usan o valoran el producto. El Product Owner (PO) anota estas sugerencias para que luego se evalúen y, si son viables, se consideren en el Product Backlog.

Ejecutantes

Product Owner

Responsable

Product Owner

Consultado

- Scrum Team, Stakeholders

Informado

- Scrum Team, Scrum Master

Descripción

Tras el análisis y cualquier acción correctiva, el Product Owner actualiza el Product Backlog para reflejar los nuevos cambios y prioridades. Esto incluye añadir nuevas tareas, ajustar el alcance y realizar cualquier ajuste necesario a las historias de usuario para alinearlas con las expectativas y las necesidades de los stakeholders.

Ejecutantes

- Product Owner

Responsable

- Product Owner

Proceso:

Actualización del Product Backlog - Actualización Del Product Backlog

5.7.1.18. Realizar documento de soporte para el lanzamiento Descripción

El Scrum Master prepara un documento de soporte que incluye toda la información relevante para el lanzamiento de la Demo. Esto puede incluir los detalles de la presentación, cualquier comentario

importante de los stakeholders y una conclusión sobre el estado actual del proyecto. Este documento será útil para la próxima planificación del sprint y como parte del material de seguimiento.

Ejecutantes

- Scrum Master

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team, Stakeholders

5.7.1.1.19. Realizar convocatoria de Planificación y estimación Descripción

El Scrum Master prepara un documento de soporte que incluye toda la información relevante para el lanzamiento de la Demo. Esto puede incluir los detalles de la presentación, cualquier comentario importante de los stakeholders y una conclusión sobre el estado actual del proyecto. Este documento será útil para la próxima planificación del sprint y como parte del material de seguimiento.

Ejecutantes

Scrum Master

Responsable

Scrum Master

Consultado

Scrum Team

Informado

Scrum Team

5.7.1.1.20. OFin del evento

Descripción

El proceso de lanzamiento (Demo) finaliza una vez que la convocatoria a la planificación y estimación es realizada. El ciclo de trabajo está listo para reiniciarse en el siguiente sprint.

5.7.1.1.21. ☐ Planeación

Descripción

Es la etapa inicial en la que se organiza el evento de lanzamiento (Demo). En esta fase se programan las fechas, se coordina la logística y se diseña la presentación que se llevará a cabo durante la Demo. El objetivo principal es asegurar que todos los preparativos estén listos para que el evento se desarrolle de forma fluida y efectiva. Esta fase se centra en la coordinación y en el diseño del contenido que se compartirá con los stakeholders y el equipo.

5.7.1.1.22. Desarrollo del evento

Descripción

La Fase de Desarrollo del Evento cubre la ejecución real del evento de lanzamiento (Demo). Esta fase se concentra en llevar a cabo la presentación de los resultados del sprint, recibir retroalimentación y evaluar si el equipo ha entregado valor. Además, se abordan posibles cambios y ajustes en el Product Backlog basados en los comentarios de los stakeholders y el Product Owner. En esta fase se cierran los ciclos de comunicación y se gestiona el avance del proyecto de forma colaborativa.

5.7.1.1.23. Documento de Observaciones

Este documento contiene una lista de observaciones, con su descripción, su categoría (Mejora, Cambio requerido, Corrección de error), su prioridad (Baja, Media, Alta), comentarios adicionales y el responsable para el seguimiento de dicha observación.

Se propone la siguiente plantilla, que se puede descargar en:

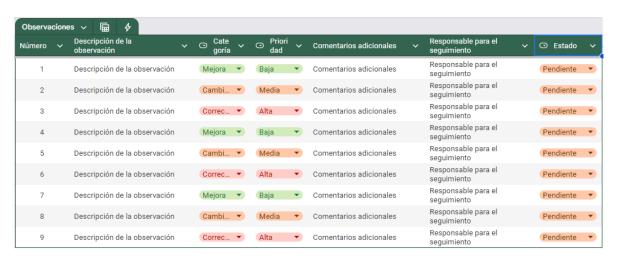


Figura 32. Plantilla de Observaciones.

5.7.1.1.24. Registro de Solicitud del Cambio

Este artefacto contiene todos los cambios sugeridos para refinar ítems en el Product Backlog por los diferentes interesados del proyecto. Contiene el ID de la solicitud, el ID de la historia de usuario a la que se le va a sugerir el cambio, el nombre descriptivo, el cambio requerido, quién fue el solicitante y cual es su rol dentro del proyecto.

Se propone la siguiente plantilla, que se puede descargar en: <u>Plantilla Registro Solicitud del Cambio</u>



Figura 33. Plantilla de Registro de solicitud del cambio.

5.7.1.1.25. Documento de Soporte de Lanzamiento

Este artefacto es desarrollado por el Scrum Master y contiene los puntos relevantes del evento como el resumen del Sprint, los detalles de la demo, el feedback recibido. el estado actual del proyecto , las conclusiones y próximos pasos que se deben tener en cuenta en próximos sprints.

Se propone la siguiente plantilla, que se puede descargar en: <u>Documento de Soporte de</u> Lanzamiento

Este artefacto contiene el listado de las historias de usuario realizadas en el sprint, con sus respectivas observaciones y un campo muy importante donde se responde si se validaron o no dichas historias de usuario.

Se propone la siguiente plantilla, que se puede descargar en:



Figura 34. Plantilla de Validación de lanzamiento.

5.7.1.2. Diseñar Presentación

Descripción

Se enfoca en la preparación de la presentación para la reunión de Lanzamiento (Demo). Este subproceso abarca desde la recopilación de la información relevante sobre el trabajo realizado en el sprint hasta la creación de los materiales visuales y la preparación técnica para la demostración del incremento del producto. La presentación debe comunicar de manera clara los avances y logros del equipo durante el sprint y demostrar el valor entregado.

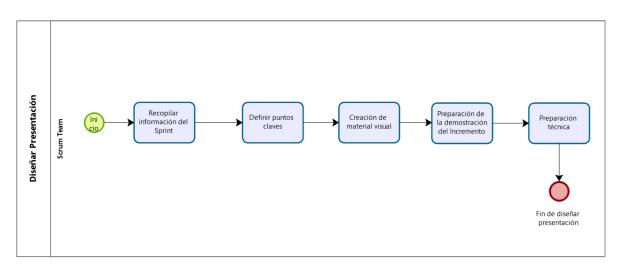




Figura 35. Proceso de Diseñar Presentación

Ejecutantes

- Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.7.1.2.1. OInicio Descripción

El proceso inicia cuando se acuerda la necesidad de preparar la presentación para el Sprint Review. En esta etapa se establece la planificación general para la presentación.

5.7.1.2.2. Recopilar información del Sprint

Descripción

En este paso, el equipo recolecta toda la información relacionada con el trabajo realizado durante el sprint. Esto incluye revisar las historias de usuario que se han completado, los elementos entregados del Product Backlog y cualquier cambio o novedad relevante en el incremento del producto.

Ejecutantes

- Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Master

Informado

- Scrum Master

5.7.1.2.3. ■ **Definir puntos claves**

Descripción

Aquí se seleccionan los puntos más relevantes del sprint que deben ser destacados en la presentación. Esto incluye las historias de usuario completadas, las funcionalidades clave, las mejoras significativas y cualquier novedad importante que deba ser resaltada ante los stakeholders.

Ejecutantes

Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Master

5.7.1.2.4. Creación de material visual

Descripción

El equipo crea el material visual necesario para la presentación, como diapositivas, gráficos, capturas de pantalla, y cualquier otro recurso visual que ayude a comunicar el progreso del sprint de forma clara y atractiva

Ejecutantes

Scrum Team

Responsable

Scrum Master

5.7.1.2.5. Preparación de la demostración del Incremento

Descripción

El equipo prepara la demostración en vivo del incremento de producto, asegurándose de que todo funcione correctamente y que se pueda mostrar de forma efectiva. Este paso incluye ensayar el flujo de la demostración y asegurarse de que se cubren todos los puntos clave.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

5.7.1.2.6. Preparación técnica

Descripción

Se realiza la configuración técnica necesaria para la demostración, lo que implica asegurar que el entorno de demostración esté completamente operativo. Esto puede incluir verificar dispositivos, conexiones a la red, entornos de desarrollo o producción, y herramientas de presentación.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

5.7.1.2.7. OFin de diseñar presentación

Descripción

El subproceso concluye cuando todo el material visual, la demostración en vivo y el entorno técnico están listos, y el equipo está preparado para presentar el Demo durante el Sprint Review.

5.7.2. Recursos

- 5.7.2.1. Scrum Master (Rol) Ya especificado en el inicio 4.1.2.2
- 5.7.2.2. Product Owner (Rol) Ya especificado en el inicio 4.1.2.1
- 5.7.2.3. Scrum Team (Rol) Ya especificado en el inicio 4.1.2.3
- 5.7.2.4. Stakeholders (Rol) Ya especificado en el inicio 4.8.2.4

5.8. Retrospectiva del sprint

Descripción

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Planificación de Sprint. Se trata de una reunión de, a lo sumo, tres horas para Sprints de un mes. Para Sprints más cortos el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito.

El Scrum Master se asegura de que la reunión sea positiva y productiva. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado. El Scrum Master participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él.

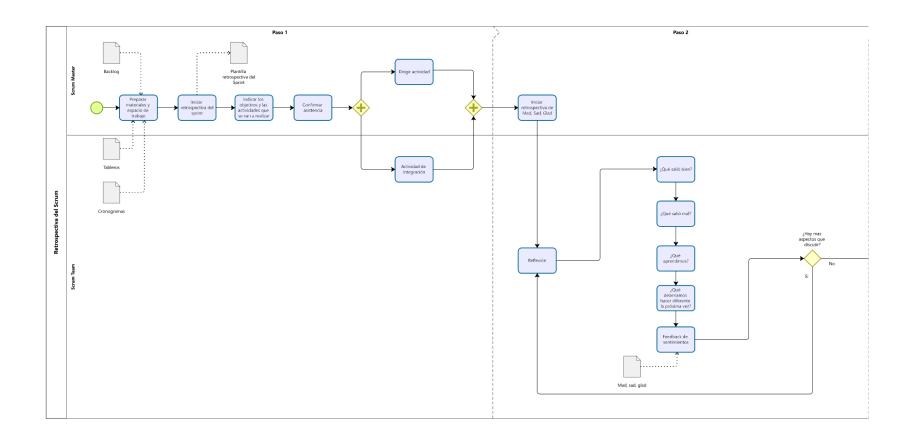


Figura 36. Proceso de Retrospectiva del Sprint.

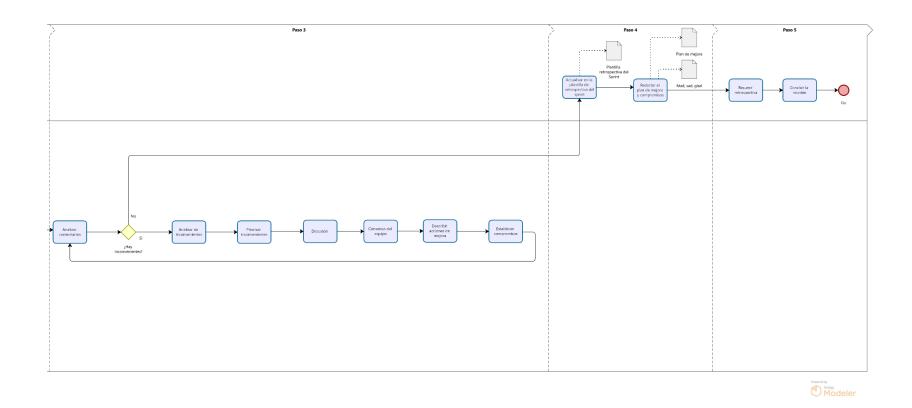


Figura 37. Proceso de Retrospectiva del Sprint Continuación.

5.8.1. Elementos del proceso

5.8.1.1. Preparar materiales y espacio de trabajo

Descripción

Se crea un ambiente adecuado para reunir los comentarios y diferentes puntos de vista para mejorar el proceso ágil. Este espacio debe ser abierto a la opinión y facilitar la generación de datos relevantes y valiosos, para este paso es fundamental contar con documentos y herramientas que permitan un análisis más integral y global del sprint. Entre los documentos y herramientas tenemos:

- Backlog
- Tableros
- Cronogramas

Ejecutantes

Scrum Master

Responsable

- Scrum Master

Informado

- Scrum Team

Descripción

La retrospectiva inicia una vez terminado el sprint es importante asegurar que la mayoría del equipo esté presente para que todos los integrantes puedan compartir su punto de vista. Se recomienda que al menos el 80% del equipo este presente, esta reunión debe realizarse de manera obligatoria al terminar el sprint, su duración depende de la duración del sprint para el caso de un sprint de un mes esta reunión debe durar máximo 3 horas, para un sprint de dos semanas aproximadamente 1hora y 30 minutos sería lo recomendable como máximo.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

Scrum Team

5.8.1.3. Indicar los objetivos y las actividades que se van a realizar

Descripción

Los objetivos deben estar dirigidos a fomentar la mejora continua, la colaboración en equipo, la resolución de problemas, evitar que se repitan los errores, mejorar la calidad y mejorar la moral del equipo. Al inicio de esta reunión se establecen objetivos claros para que todos los participantes estén dispuestos a colaborar de manera efectiva en esta reunión.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.4. Confirmar asistencia

Descripción

Se verifica que se encuentren los integrantes del equipo, el product owner y los interesados.

Es de vital importancia tratar de que el equipo de scrum se encuentre presente ya que el éxito del proyecto depende de cada uno de ellos, cada miembro nos puede aportar puntos de vista diferentes para mejorar la calidad del producto y del proceso, además es importante que se encuentre la mayoría de los integrantes para llegar a un acuerdo en conjunto y unos objetivos y metas claras y disposición de comprometerse con las mejoras propuestas.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.5. Dirigir actividad

Descripción

Asegurarse de que la actividad de integración siga los lineamientos y no exceda el tiempo previsto, esto con el fin de garantizar que se obtengan los resultados esperados en la planeación.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.6. Actividad de integración

Descripción

Esta actividad puede ser dirigida ya sea por el scrum máster o también por un experto en relaciones interpersonales, según las necesidades del equipo, el propósito principal de esta actividad es fortalecer la confianza y la cohesión entre los miembros del equipo, creando así un ambiente seguro y abierto para compartir sus experiencias y expresar cualquier inconveniente y preocupación sin temor.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Master

Informado

- Scrum Team

Descripción

El scrum master sienta las bases de la sesión indicando que este tiempo es para hablar de como se ha trabajado y como se puede mejorar, se da por supuesto que todo mundo dio lo mejor de sí, que nada de lo que se comparta será usado contra nadie y que el espacio es para reflexionar y no para culpar.

En este caso se aplica una técnica llamada MAD GLAD SAD que nos ayuda a expresar las emociones y experiencias en tres categorías distintas.

Ejecutantes

Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.8. ■ Reflexión

Descripción

Los integrantes del equipo deben pensar cuáles fueron los acontecimientos clave, en los siguientes pasos se va a descomponer esta reflexión en 5 etapas.

Ejecutantes

Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.9. □¿Qué salió bien?

Descripción

En esta etapa se requiere reflexionar aspectos positivos del sprint, se identifican acciones que funcionaron de manera correcta, los logros alcanzados que acciones hicieron que se cumplan las metas.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.10. Analizar comentarios

Descripción

En esta etapa se busca un análisis exhaustivo de los comentarios identificados para revisarse detalladamente, identificar los problemas que puedan impedir mejoras, pero también resaltar los comentarios positivos para mantenerlos y que no se conviertan en mejoras a largo plazo.

Ejecutantes

- Scrum Team

Responsable

- Scrum Team

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.11. Analizar de inconvenientes

Descripción

El equipo de desarrollo se reúne con el Scrum Master para revisar todo tipo de inconvenientes encontrados. Para esto se documentarán aspectos que diferencien el tipo de inconveniente, la frecuencia con la que ocurrió, el impacto que causó en el progreso y las causas que posiblemente lo causaron desde la raíz. También se tendrá un análisis colaborativo que ayude a determinar áreas que necesiten ajustes.

Ejecutantes

Scrum Team

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.12. Priorizar inconvenientes

Descripción

Durante la retrospectiva, el equipo junto con el Scrum Master revisarán cada uno de los inconvenientes identificados en el sprint finalizado y se hará una clasificación de acuerdo con la gravedad, frecuencia e impacto en el cumplimiento de los propósitos del sprint. Para esto podría incluirse una matriz que clasifique la prioridad del inconveniente (alta, media y baja). Esto con el propósito de ayudar al equipo a enfocar los esfuerzos en resolver los problemas más críticos de manera rápida pero estructurada.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.13. Discusión

Descripción

Para la discusión acerca de los inconvenientes el equipo junto con el Scrum Master se reunirá para discutir los inconvenientes priorizados donde revisarán las causas, el impacto y las posibles soluciones. Con esto se fomenta a que cada miembro del equipo comparta su forma de verlo personalmente junto con las experiencias y proponiendo alternativas para la resolución de estos inconvenientes, donde se podrán mejoras en los procesos o prácticas nuevas que puedan implementarse. El papel del Scrum Master es fundamental en esta tarea ya que ayudará a facilitar la conversación para asegurarse de que todos los puntos de vista expuestos sean tomados en cuenta y que las ideas para el plan de mejora sean alcanzables.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.14. Consenso del equipo

Descripción

El equipo se reúne para revisar las propuestas de solución planteadas y discutidas previamente para llegar a un consenso sobre cuáles acciones se implementarán. Se invitará a cada miembro del equipo a expresar su opinión y votar o apoyar estas acciones que según su criterio considere más efectivas, pero sobre todo viables. Durante esta tarea, el Scrum Máster facilitará el consenso asegurando que las decisiones sean consecuentes y alineadas con las capacidades del equipo y con los objetivos propuestos.

Ejecutantes

- Scrum Master, Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.15. Describir acciones de mejora

Descripción

Basándose en el consenso alcanzado, el equipo se dedicará a definir con precisión las acciones de mejora identificadas. Esto incluye la descripción detallada de cada acción, los pasos a seguir, los recursos necesarios y los miembros responsables de cada implementación. Es importante que cada una de las acciones esté formulada de manera clara, y que refleje los resultados para facilitar evaluarlos en próximos Sprints.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.16. Establecer compromisos

Descripción

Después de definir y detallar las acciones de mejora, el equipo dedicará un momento para poder establecer compromisos específicos. Cada miembro deberá expresar su disposición a asumir la responsabilidad de ciertas acciones y el cumplimiento de las tareas asignadas dentro de los tiempos estimados y establecidos. El Scrum Master ayudará a facilitar esta actividad, siendo este el que promueva la actitud de apoyo mutuo y un ambiente en el que el compromiso, aunque sea individual fortalezca el esfuerzo colectivo hacía la mejora continua.

Ejecutantes

- Scrum Master, Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.17. Actualizar en la plantilla de retrospectiva del sprint

Descripción

Consiste en incorporar las nuevas tareas o ajustes acordados a la plantilla de la retrospectiva del sprint. El objetivo es mantener esta actualizada y alineada con las prácticas y necesidades del equipo, donde se debe asegurar que todas las etapas están siempre claras en su descripción y orden. Es importante establecer que los objetivos cumplan con SMART (Specific-Measurable-Achievable-Relevant).

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.18. Redactar el plan de mejora y compromisos

Descripción

Se crea un documento que detalla las acciones de mejora acordadas durante la retrospectiva, también los compromisos asumidos por cada miembro del equipo. En este plan de mejora se debe incluir la descripción de manera clara de cada acción, los responsables, los plazos acordados, y cualquier recurso o apoyo que se considere necesario. Además, se deben establecer métricas o criterios de éxito para evaluar el impacto de cada acción en el próximo sprint. Este documento será una referencia clara y compartida que ayudará a guiar al equipo en la implementación, y promoverá la responsabilidad y el seguimiento en las retrospectivas futuras.

Ejecutantes

Scrum Master

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.19. Resumir retrospectiva

Descripción

Esta tarea consiste en sintetizar los puntos clave tratados en la retrospectiva, donde se deben incluir los inconvenientes, las acciones de mejora, los compromisos que se asumieron por el equipo y cualquier otro aprendizaje significativo. Este resumen debe ser claro, conciso y sobre todo deben verse reflejadas las decisiones y acuerdos alcanzados, que proporcione un registro de referencia para el equipo. Este documento permitirá hacer un seguimiento fácil sobre las mejoras en los sprints futuros y también asegura que todos los miembros tengan claridad sobre todos los pasos a seguir.

Ejecutantes

- Scrum Master

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.20. Concluir la reunión

Descripción

Marca el cierre de la retrospectiva. El scrum master da un breve resumen de los compromisos y acuerdos alcanzados, no sin antes agradecer las contribuciones de cada miembro. Además, se invita al equipo a compartir cualquier comentario final o extra sobre la retrospectiva. Esto ayuda a consolidar el sentido de unidad y compromiso del equipo.

Ejecutantes

- Scrum Master

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.21. □¿Qué salió mal?

Descripción

Aquí se analiza qué problemas o dificultades se presentaron en el sprint que cosas influyeron en retrasos o errores técnicos o problemas de comunicación entre el equipo. Se trata de buscar los obstáculos que se presentaron en el desarrollo del sprint, se debe destacar que este paso es muy importante para poder comprender los problemas de una mejor manera y poder plantear futuras soluciones.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.22. □¿Qué aprendimos?

Descripción

En esta sección se identifican las actividades y las lecciones o procesos que nos parecieron enriquecedores o que puedan aportar al equipo que nos permitan mejorar y adaptarse de mejor manera en el futuro.

Ejecutantes

Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.23. Qué deberíamos hacer diferente la próxima vez?

Descripción

En esta etapa el equipo identifica etapas o partes específicas que requieren mejoras, se trata de identificar partes del proceso específicas para evitar futuros errores o problemas anteriores se identifica en base a experiencias vividas en el transcurso del sprint.

Ejecutantes

- Scrum Team

Responsable

- Scrum Master

Consultado

Scrum Team

Informado

- Scrum Team

5.8.1.24. Feedback de sentimientos

Descripción

Finalmente, para terminar esta sección de la retrospectiva tenemos el feedback de sentimientos lo ideal sería que una persona experta en el tema realice este análisis, se trata de que todos los miembros compartan sus emociones durante el sprint esto nos ayuda a comprender mejor la dinámica del equipo y poder crear un ambiente donde todos se sientan escuchados y comprendidos y poder también en base a sus sentimientos acciones de mejora para que el ánimo del equipo permanezca de la mejor manera.

Ejecutantes

- Scrum Team

Responsable

Scrum Master

Consultado

- Scrum Team

Informado

- Scrum Team

5.8.1.25. Plantilla retrospectiva del Sprint

Descripción

Sirve para fomentar la mejora continúa, fortalecer la colaboración, identificar éxitos y aprendizajes, gestionar problemas y establecer compromisos.

Este se conforma de una introducción y objetivo de la reunión y un resumen del sprint donde se tienen preguntas clave como, qué salió bien, mal, qué podemos mejorar, seguido de ideas de mejora, lecciones aprendidas, plan de acción y responsables.

Plantilla retrospectiva del sprint

5.8.1.26. Mad, sad, glad

Descripción

MAD (Enojado): Aquí cada miembro del equipo comparte qué cosas lo hicieron sentir enojado durante el sprint.

GLAD (Contento): Aquí cada miembro del equipo comparte qué cosas lo hicieron sentir contento durante el sprint, incluyendo el apoyo con los compañeros o que lo hizo sentir satisfecho.

SAD (Triste): En este punto, el equipo habla de lo que les causó tristeza o decepción. Pueden ser situaciones que no salieron como esperaban o problemas que afectaron la moral del equipo.

Plantilla mad, sad, glad

5.8.1.27. Plan de mejora

Descripción

La plantilla de plan de mejora es útil para documentar y organizar acciones enfocadas en mejoras de procesos, prácticas o resultados específicos. Con el fin de resolver problemas y optimizar el rendimiento del equipo y también del proyecto.

Este nos sirve para identificar y estructurar mejoras necesarias, definir acciones específicas y medibles, asignar responsabilidades, evaluar el progreso y fomentar la mejora continúa.

Plantilla de retrospectiva del sprint y plan de mejora

5.8.1.28. Tableros

Descripción

Los tableros de Jira te permiten dividir un proyecto grande en partes más manejables para que puedas planificar tus sprints y, de este modo, agilizar tus entregas.

Tableros

Plantilla tableros de sprint

5.8.1.29. **Backlog**

Descripción

El backlog de un producto es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de su hoja de ruta y sus requisitos. Los elementos más importantes se muestran al principio del backlog del producto para que el equipo sepa qué hay que entregar primero.

5.8.1.30. Cronogramas

Descripción

Los cronogramas mantienen a tu equipo en sincronía con las partes interesadas y proporcionan una visión de conjunto del cronograma del proyecto con los elementos de trabajo, dependencias y publicaciones asignados.

5.8.1.31. Paso 1

Descripción

Establecer el contexto

5.8.1.32. Paso 2

Descripción

Recopilar comentarios

5.8.1.33. Paso 3

Descripción

Generar datos relevantes

5.8.1.34. Paso 4

Descripción

Crear elementos de acción

5.8.1.35. Paso 5

Descripción

Concluir la retrospectiva

5.8.2. RECURSOS

5.8.2.1. Scrum Master (Rol)

El Scrum Master tiene como funciones y responsabilidades:

- -Facilitar las reuniones de scrum como, por ejemplo: planning, daily, sprint reviews, retrospectives, asegurándose que haya un enfoque en sus objetivos y que el equipo mantenga el ritmo.
- -Eliminar impedimentos: identificándolos y sacándolos para que no haya impedimentos en el progreso del equipo.
- -Fomentar la mejora continua: Ayudar al equipo a evaluar los procesos y a identificar cualquier oportunidad de mejora (preferiblemente en las retrospectivas).
- -Promover autonomía y el trabajo en equipo: apoyándolos para que tomen decisiones y se adueñan de sus tareas, fomentando un entorno donde haya confianza.
- -Mediar entre Stakeholders y el equipo de desarrollo: Para que los interesados externos comprendan el proceso scrum y las necesidades a suplir.

Sin pasar por alto que este debe tener la habilidad blanda de comunicarse fácilmente, ser proactivo y resolutivo en conflictos.

5.8.2.2. Scrum Team (Rol)

Scrum team es un grupo multifuncional y autogestionado que trabaja en conjunto para entregar productos de valor, donde se tiene el product owner, scrum master y development team.

5.9. Reunión de Gestión del Cambio

Descripción

La reunión de gestión del cambio se lleva a cabo para identificar y discutir ajustes necesarios en las tareas del sprint actual, particularmente si surgen riesgos o si se observa una desviación inesperada en el proceso. Esta reunión puede integrarse en la reunión de refinamiento (grooming) del backlog y está orientada a gestionar cambios en los ítems ya existentes en el Sprint Backlog, sin introducir nuevos elementos.

5.9.1. Elementos del proceso

5.9.1.1. CLista de cambios aprobados está vacía ?

Descripción

El Product Owner verifica si la lista de cambios aprobados está vacía. Si la lista se encuentra vacía significa que no se agregaran cambios al producto backlog por lo que se dará por terminado el proceso, en caso contrario se continuará con las siguientes etapas.

Flujos

- NO
- SI

5.9.1.2. Refinamiento del Backlog

Descripción

El Product Owner refina el Product Backlog para incluir los cambios aprobados, asegurando que estén bien definidos y priorizados. Esto puede implicar descomponer cambios en tareas más pequeñas y claras para el Development Team.

Ejecutantes

Product Owner

Responsable

Product Owner

Consultado

- Development Team

Informado

- Scrum Master

5.9.1.3. Comienzo del Proceso de Gestión del Cambio

Descripción

Este es el punto inicial del proceso.

5.9.1.4. Preparar materiales y espacio de trabajo

Descripción

El Scrum Máster se encarga de preparar el espacio donde se llevará a cabo la reunión, también los materiales necesarios, como pizarras, marcadores y cualquier herramienta necesaria para presentar y discutir los cambios, así como garantizar que la lista de asistencia y la agenda de la reunión estén listas.

Ejecutantes

Scrum Master

Responsable

- Scrum Master

Consultado

- Product Owner

Informado

- Development Team

5.9.1.5. Iniciar reunión de gestión del cambio

Descripción

El Scrum Máster inicia la reunión y procede a realizar una lista de asistencia para ver los miembros que han asistido a la reunión.

Ejecutantes

Scrum Master

Responsable

Scrum Master

Consultado

- Product Owner, Development Team

Informado

Stakeholders

5.9.1.6. Definición final de tareas

Descripción

El Development Team define en detalle las tareas específicas para implementar los cambios, incluyendo subtareas, dependencias y cualquier requerimiento técnico adicional.

Ejecutantes

Development Team

Responsable

Product Owner

Consultado

- Scrum Master

5.9.1.7. Planificación y Estimación

Descripción

La planificación de sprint es el evento Scrum diseñado para asegurarse de que el equipo esté preparado para hacer las cosas correctas en cada sprint.

La duración de la mayoría de los eventos de Scrum está relacionada con la duración del sprint. En cuanto a la planificación de sprint, debe durar 2 veces la duración del sprint (en horas). Por ejemplo, si tu sprint dura 2 semanas, la reunión de planificación de sprint no debe durar más de 4 horas. Para un sprint de 1 semana, no debe durar más de 2 horas.

Ejecutantes

Development Team

Responsable

Product Owner

Consultado

- Scrum Master

5.9.1.8. OFinalización del Proceso de Gestión del Cambio

Descripción

Este es el punto de conclusión del proceso.

5.9.1.9. Lista de asistencia

Descripción

Registro de los participantes en la reunión de gestión de cambios. Este documento asegura que todos los interesados relevantes estuvieron presentes y participaron en la discusión.

5.9.1.10. Lista de cambios propuestos

Descripción

Documento que contiene todos los cambios identificados y propuestos para revisión. Incluye una breve descripción de cada cambio y su justificación inicial.

5.9.1.11. Dustificación de no aprobado

Descripción

Documento en el que se registran las razones y argumentos por los cuales un cambio propuesto no fue aprobado. Esto proporciona transparencia y rastreabilidad en el proceso de toma de decisiones.

5.9.1.12. Lista de cambios aprobados

Descripción

Listado de todos los cambios que han sido revisados y aprobados para su implementación.

5.9.1.13. Lista de tareas refinadas

Descripción

Documento en el que se detallan las tareas específicas derivadas de los cambios aprobados.

5.9.1.14. Descripción detallada de cada tarea

Descripción

Documento o registro en el que se detalla cada tarea específica que será llevada a cabo para implementar los cambios. Incluye objetivos, subtareas, dependencias y cualquier especificación técnica o funcional.

5.9.1.15. Product Backlog

Descripción

Versión del Product Backlog que incorpora todas las tareas refinadas y los cambios aprobados. Esta versión no contiene estimación de tiempo y esfuerzo.

Descripción

El Development Team identifica todos los cambios potenciales que podrían agregarse al Product Backlog. Estos cambios pueden incluir la adición de nuevas funcionalidades, la eliminación de funcionalidades existentes o la modificación de funcionalidades ya implementadas.

Ejecutantes

- Development Team

Responsable

- Product Owner

Consultado

- Scrum Master, Stakeholders

Informado

Development Team

5.9.1.17. Socialización de los cambios

Descripción

El Product Owner agrega detalles a cada tarea para poder ser comunicados de manera correcta. Después de agregados los detalles se procede a comunicar los cambios aprobados al Development Team y a otros interesados relevantes para garantizar una comprensión compartida. Esta comunicación ayuda a alinear expectativas y prepara al equipo para implementar los cambios en próximos sprints.

Ejecutantes

- Product Owner

Responsable

- Product Owner

Consultado

- Scrum Master, Development Team

Informado

- Stakeholders

Descripción

El Product Owner toma la lista de cambios propuestos y procede a iterar en los cambios que hay en ella para poder evaluar los que serán viables para el Product Backlog.

Ejecutantes

- Product Owner

Responsable

- Product Owner

Consultado

- Scrum Master, Development Team

5.9.1.19. Descripción

El Product Owner verifica si aún quedan cambios en la lista para revisar. Si la lista está vacía, se pasa al siguiente paso; de lo contrario, se continúa revisando cada cambio.

Flujos

- NO
- SI

5.9.1.20. Seleccionar un cambio de la lista

Descripción

El Product Owner selecciona un cambio específico de la lista para su evaluación. Este cambio se retira de la lista de cambios propuestos y se procede a ser evaluado.

Ejecutantes

Product Owner

Responsable

Product Owner

Consultado

Development Team

☐ Finalizar iteración de posibles cambios propuestos 5.9.1.21.

Descripción

Marca el cierre de la revisión de la lista de cambios propuestos.

Ejecutantes

Product Owner

Responsable

Product Owner

Consultado

Scrum Master, Development Team

5.9.1.22.

Descripción

El Product Owner verifica la decisión tomada con el posible cambio. Si el cambio es aprobado se pasará a las etapas de aprobación, en caso contrario se pasará a la etapa de desaprobación.

Flujos

- NO
- SI

5.9.1.23. Evaluación del impacto

Descripción

El equipo de desarrollo y el Product Owner evalúan el impacto del cambio en el backlog, en el sprint actual (si aplica), y en el valor del producto final. Esto incluye analizar los recursos y tiempo necesarios para implementar el cambio. Si el cambio es de tipo "modificar" o "eliminar," se revisan los efectos en funcionalidades dependientes.

Ejecutantes

- Development Team

Responsable

- Product Owner

Consultado

- Scrum Master

5.9.1.24. Ustificar la desaprobación

Descripción

Si un cambio no es aprobado, el Product Owner documenta las razones y justifica la decisión para mantener una comunicación clara con los interesados. Esto ayuda a asegurar transparencia y a documentar las decisiones para futuras revisiones.

Ejecutantes

- Product Owner

Responsable

- Product Owner

Consultado

- Scrum Master

Informado

- Development Team

5.9.1.25. Agregar a la lista de cambios aprobados Descripción

Si el cambio es aprobado, clasificado y evaluado, se añade a la lista de cambios aprobados. Esta lista representa todas las modificaciones que se incorporarán al Product Backlog para su posterior refinamiento y planificación.

Ejecutantes

- Product Owner

Responsable

Product Owner

Consultado

Development Team

Informado

- Scrum Master

Descripción

El Product Owner clasifica el cambio en función de su tipo: añadir (nueva funcionalidad o característica), eliminar (quitar una funcionalidad obsoleta o no prioritaria), o modificar (cambiar una funcionalidad existente). Esta clasificación es crucial ya que determina las acciones subsecuentes, como el nivel de impacto y el esfuerzo de desarrollo.

Ejecutantes

Product Owner

Responsable

- Product Owner

Consultado

- Development Team, Scrum Master

5.9.2. RECURSOS

5.9.2.1. Product Owner (Rol)

Ya especificado en Product Owner (Rol).

5.9.2.2. Development Team (Rol)

El Development Team es un grupo multifuncional de profesionales que se encargan de construir el producto. Este equipo es autónomo y autogestionado, lo que significa que decide cómo realizar su trabajo y se organiza de la manera que considera más efectiva. Los miembros del equipo pueden tener habilidades específicas (por ejemplo, desarrollo, pruebas, diseño), pero trabajan juntos como un grupo cohesivo. En Scrum, el equipo de desarrollo tiene el objetivo de crear un incremento del producto que esté listo para ser entregado al final de cada Sprint.

5.9.2.3. Scrum Master (Rol)

Ya especificado en Scrum Master (Rol).

5.9.2.4. Stakeholders (Rol)

Los stakeholders son personas externas al equipo Scrum, como clientes, usuarios, gerentes, u otras partes interesadas que tienen un interés en el producto. Aunque no participan directamente en el proceso de desarrollo, los stakeholders proporcionan retroalimentación y ayudan a definir los requisitos y expectativas del producto.

6. Conclusiones

La realización de este proceso permitió formalizar y documentar un enfoque estructurado basado en Scrum, adaptado a las necesidades de micro, pequeñas y medianas empresas de desarrollo de software. Este proyecto demostró que, a través de un análisis detallado de los elementos clave de Scrum, es posible diseñar un manual que brinde guías claras y prácticas para su implementación. Esta formalización contribuye a superar las barreras que enfrentan muchas organizaciones al adoptar metodologías ágiles, especialmente la falta de un proceso documentado que contemple tareas específicas, roles definidos y actividades complementarias.

El desarrollo del proyecto no solo resultó en la creación de un recurso útil para las organizaciones, sino que también potenció el aprendizaje de los integrantes del grupo. La experiencia permitió una comprensión profunda de los principios y prácticas de Scrum, fortaleciendo habilidades como la planificación, la organización y el trabajo colaborativo en entornos ágiles. Asimismo, la utilización de Bizagi como herramienta para modelar y detallar el proceso fue clave para representar de manera visual y estructurada cada fase, actividad y evento del marco de trabajo. Este enfoque destacó la importancia de contar con herramientas adecuadas para apoyar la documentación y gestión de procesos ágiles.

Finalmente, este proyecto no solo dejó un recurso práctico que facilita la adopción de Scrum en las empresas, sino que también enfatizó la relevancia de los principios fundamentales de esta metodología, como la autoorganización, la colaboración y la responsabilidad de los equipos. De esta manera, se creó una solución integral que no solo mejora la capacidad de las empresas para implementar Scrum, sino que también refuerza los valores y las prácticas que lo hacen efectivo.

7. Bibliografía

- [1] "Características, historias y tareas iceScrum." Accessed: Nov. 13, 2024. [Online]. Available: https://www.icescrum.com/es/documentation/features-stories-tasks/
- [2] "Epic Scrum Manager BoK." Accessed: Nov. 13, 2024. [Online]. Available: https://www.scrummanager.com/bok/index.php/Epic
- [3] "6 marcos de priorización de productos [+Cómo elegir]." Accessed: Nov. 13, 2024. [Online]. Available: https://www.atlassian.com/es/agile/product-management/prioritization-framework
- [4] "Qué es la priorización y cómo se utiliza en el desarrollo de aplicaciones Apliint Software Development." Accessed: Nov. 13, 2024. [Online]. Available: https://apliint.com/2022/03/01/que-es-la-priorizacion-y-como-se-utiliza-en-el-desarrollo-de-aplicacio nes/
- [5] "Historias de usuario | Ejemplos y plantilla | Atlassian." Accessed: Nov. 13, 2024. [Online]. Available: https://www.atlassian.com/es/agile/project-management/user-stories
- [6] "Explicación del backlog del producto [+ ejemplos]." Accessed: Nov. 13, 2024. [Online]. Available: https://www.atlassian.com/es/agile/scrum/backlogs
- [7] "Qué es el product backlog y guía para hacer uno con ejemplo [2024] Asana." Accessed: Nov. 13, 2024. [Online]. Available: https://asana.com/es/resources/product-backlog
- [8] "Qué es un product backlog y cómo hacer uno [Guía Scrum]." Accessed: Nov. 13, 2024. [Online]. Available: https://www.iebschool.com/blog/que-es-un-product-backlog-y-como-hacer-uno-guia-scrum-agile-scru m/
- [9] D. Molina, "Qué es un product backlog y cómo hacer uno [Guía Scrum]," Thinking for Innovation, Jun. 2023, Accessed: Nov. 13, 2024. [Online]. Available: https://www.iebschool.com/blog/que-es-un-product-backlog-y-como-hacer-uno-guia-scrum-agile-scru m/
- [10] "Plantillas Scrum: Pila de producto (Product Backlog) La Oficina de Proyectos de Informática." Accessed: Nov. 13, 2024. [Online]. Available: https://www.pmoinformatica.com/2013/11/plantillas-scrum-pila-producto-product.html
- [11] "Scrum: ¿Qué es el Product Backlog?" Accessed: Nov. 13, 2024. [Online]. Available: https://programacionymas.com/blog/scrum-product-backlog
- [12] G. Wagenaar, "What is Ready in a DoR? Rationales, Responsibility, Rules in using a Definition of Ready," ResearchGate, Nov. 2024. [Online]. Available: https://www.researchgate.net/publication/381164049_What_is_Ready_in_a_DoR_Rationales_Respon sibility_Rules_in_using_a_Definition_of_Ready. [Accessed: Nov. 7, 2024].
- [13] T. Versteeg, "What is Ready in a DoR? Rationales, Responsibility, Rules in using a Definition of Ready," Master's thesis, Utrecht University, 2024. [Online]. Available: https://studenttheses.uu.nl/bitstream/handle/20.500.12932/44214/Final_MBI_Thesis_DoR_TommyVe rsteeg.pdf?sequence=1&isAllowed=y. [Accessed: Nov. 7, 2024].

- [14] Atlassian, "Sprint Reviews," Atlassian, [Online]. Available: https://www.atlassian.com/es/agile/scrum/sprint-reviews. [Accessed: Nov. 15, 2024].
- [15] K. Schwaber y J. Sutherland, "Guía de Scrum: La Guía Definitiva de Scrum: Las Reglas del Juego," Scrum.org, Nov. 2017. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf. [Accessed: Nov. 15, 2024].
- [16] K. Schwaber y J. Sutherland, "Sprint Planning," Scrum.org, Nov. 2017. [Online]. Available: https://scrumguides.org/scrum-guide.html#sprint-planning. [Accessed: Nov. 15, 2024].
- [17] J. Palacios, "Guía al movimiento #NoEstimates," Jeronimo Palacios, [Online]. Available: https://jeronimopalacios.com/agile/guia-al-movimiento-noestimates/. [Accessed: Nov. 17, 2024].