

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



Modelo Computacional de equipos de trabajo en organizaciones
con un enfoque de Sistemas Viables

TESIS

QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS

Presenta:

ERNESTO RAÚL ALVAREZ MOLINA

Director de tesis:

DR. LUIS GUILLERMO MARTÍNEZ MÉNDEZ

Codirector de tesis:

DR. ANTONIO RODRÍGUEZ DÍAZ

TIJUANA, BAJA CALIFORNIA, MÉXICO

DICIEMBRE 2017

Dedicatoria

A Dios

Por ser la fuerza e inspiración que me mueve en todo momento.

A Mis padres

Por creer en mí y brindarme su apoyo incondicional a lo largo de mi vida.

Agradecimientos

A mi Director , Coodirector y resto del equipo

Por su valiosa ayuda en la realización de uno de mis proyectos de vida.

A mis Maestros

Por su compartirme sus valiosos conocimientos y experiencia.

A mis Amigos

Por haber hecho mas a meno el trayecto hacia esta importante meta.

A conacyt

Por el apoyo otorgado para la realizacion de este proyecto.

Ernesto Raúl Álvarez Molina

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 227

Tijuana, B. C., a 23 de noviembre de 2017

C. Ernesto Raúl Álvarez Molina
Pasante de: Doctor en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la

Opción TESIS

Es propuesto, por los C. Dres. Luis Guillermo Martínez Méndez y Antonio

Rodríguez Díaz

Quienes serán los responsables de la calidad de trabajo que usted presente, referido al tema Modelo Computacional de equipos de trabajo en organizaciones con un enfoque de Sistemas Viables.

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- MARCO TEÓRICO
- III.- MODELO DE SISTEMA VIABLE DE UN EQUIPO SCRUM
- IV.- MODELO BASADO EN AGENTES DE UN EQUIPO SCRUM
- V.- CASO DE ESTUDIO
- VI.- RESULTADOS
- VII.- CONCLUSIONES

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA


Dr. Luis Guillermo Martínez Méndez
Director de Tesis


Dr. José Luis González Vázquez
Sub-Director Secretario


Dr. Antonio Rodríguez Díaz
Co-Director de Tesis


Dr. Luis Enrique Palafox Maestre
Director

RESUMEN

Los equipos de trabajo son un importante recurso para las organizaciones; y en las que se dedican al desarrollo de software los equipos SCRUM son un claro ejemplo de los tipos de equipo que han aumentado actualmente con el auge del uso de los métodos ágiles en el proceso de desarrollo de software. La necesidad por parte de las organizaciones de poder contar con equipos efectivos que logren los objetivos de manera efectiva y eficiente ha vuelto el área del trabajo en equipo un tópico de interés para la investigación científica. En este trabajo de investigación se propone un modelo computacional enfocado en el Modelo de Sistemas Viables que aborda aspectos organizacionales en el marco de trabajo de los equipos SCRUM, además de los aspectos individuales de los integrantes desde el punto de vista del Modelo de Sistemas Viables. Se tomó como caso de estudio en equipo de desarrollo de SCRUM para implementarlo en un simulador generando un prototipo computacional que aborda el aspecto de la asignación de tareas en equipos SCRUM comparando tres escenarios distintos. Dicha implementación es el primer paso hacia una herramienta integral de un laboratorio virtual de equipos SCRUM.

ABSTRACT

Teams are an important resource for organizations; the SCRUM teams are a clear example in the domain of software development, proof of this is the current boom in the use of agile methods such as the case of the SCRUM framework. The needs of organizations to count with effective teams have turned the area of teamwork into a topic of interest for scientific research, helping them to achieve their objectives effectively and efficiently. In this research work we propose a computational model focused on the Viable Systems Model that addresses organizational aspects of the SCRUM framework, as well as the individual aspects of the members of a team, from the point of view of the Viable System Model. An implementation of the model was carried out in a case study by means of a simulator prototype that addresses the aspect of assignment of tasks in a SCRUM team; with this prototype three different scenarios were executed. This implementation is the first step towards an integral tool of a virtual laboratory of SCRUM teams.

Índice general

CAPÍTULO 1 INTRODUCCIÓN.....	8
1.1 MOTIVACIÓN	8
1.2 ANTECEDENTES	9
1.3 JUSTIFICACIÓN	12
1.4 PLANTEAMIENTO DEL PROBLEMA.....	12
1.5 OBJETIVO GENERAL	14
1.6 OBJETIVOS ESPECÍFICOS	14
1.7 ESTRUCTURA DE LA TESIS.....	14
CAPÍTULO 2 MARCO TEÓRICO.....	15
2.1 MODELO DE SISTEMAS VIABLES.....	15
2.1.1 Subsistemas:	16
2.1.1.1 Sistema 1 Operación:	16
2.1.1.2 Sistema 2 Coordinación.....	16
2.1.1.3 Sistema 3* Auditoria.....	19
2.1.1.4 Sistema 4 Inteligencia	20
2.1.1.5 Sistema 5 Política.....	21
2.2 MODELADO BASADO EN AGENTES Y SIMULACIÓN.....	23
2.3 METODOLOGÍA ÁGIL SCRUM.....	24
2.3.1.1 Dueño del producto (PO)	27
2.3.1.2 Equipo de Desarrollo (DT) o Miembro del Equipo (TM)	27
2.3.1.3 Maestro SCRUM (SM)	28
CAPÍTULO 3 MODELO DE SISTEMA VIABLE DE UN EQUIPO SCRUM.....	33
3.1 IDENTIDAD DEL SISTEMA.....	33
3.1.1 Entorno presente y futuro.....	35
3.2 DIMENSIÓN VERTICAL	36
3.2.1 Desdoblamiento de la complejidad: criterios y niveles de recursión	36
3.3 DIMENSIÓN HORIZONTAL	37
3.3.2.1 Sistema 5 Política.....	38
3.3.2.2 Sistema 4 Inteligencia	41
3.3.2.3 Sistema 3 Integración.....	42
3.3.2.4 Sistema 3* Auditoria.....	43
3.3.2.5 Sistema 2 Coordinación.....	43
3.3.2.6 Sistema 1 Operación	45
3.4 COHERENCIA DE LA RECURSIÓN DEL SISTEMA.....	45
CAPÍTULO 4 MODELO BASADO EN AGENTES DE UN EQUIPO SCRUM	48
CAPÍTULO 5 CASO DE ESTUDIO	54
CAPÍTULO 6 RESULTADOS.....	57
CAPÍTULO 7 CONCLUSIONES	65
 BIBLIOGRAFIA	 67
ANEXO 1 ENCUESTA PROTOTIPO	70

Índice de figuras

FIGURA 1 MODELO DE SISTEMA VIABLE ADAPTADO DE [15].....	15
FIGURA 2 SISTEMA UNO	16
FIGURA 3 SISTEMA 2 EN COLOR GRIS.....	17
FIGURA 4 SISTEMA 3 EN COLOR GRIS.....	18
FIGURA 5 RELACIÓN DEL SISTEMA TRES CON CADA UNIDAD OPERATIVA.....	19
FIGURA 6 SISTEMA TRES* EN COLOR GRIS	19
FIGURA 7 SISTEMA 4 INTELIGENCIA	20
FIGURA 8 SISTEMA 5 POLÍTICA	21
FIGURA 9 ESQUEMA GENERAL DE UN ABMS ADAPTADO DE (SAYAMA, 2015).....	23
FIGURA 10 MARCO DE TRABAJO SCRUM.....	25
FIGURA 11 COMPOSICIÓN DE UN EQUIPO SCRUM.....	26
FIGURA 12 META DEL SPRINT	29
FIGURA 13 ENTRADAS Y SALIDAS DE LA PLANEACIÓN DEL SPRINT	30
FIGURA 14 PILA DEL PRODUCTO	32
FIGURA 15 PROPÓSITO Y LÍMITES DE UN EQUIPO SCRUM.....	34
FIGURA 16 DIMENSIÓN VERTICAL: SISTEMA FOCAL Y CRITERIO DE RECURSIÓN.....	36
FIGURA 17 MODELO DE SISTEMA VIABLE DE UN EQUIPO DE DESARROLLO SCRUM	37
FIGURA 18 ESFERA DE RECURSIÓN DEL CRITERIO: PROYECTO	46
FIGURA 19 VSM Y ABMS	48
FIGURA 20 PROCESO DEL VSM A ABM.....	49
FIGURA 21 INTERACCIONES EN LA DINÁMICA DE SCRUM.....	51
FIGURA 22 DIAGRAMA DE CLASES DE SCRUM.....	53
FIGURA 23 INTERFACE DEL SIMULADOR EN NETLOGO	54
FIGURA 24 ZONAS DEL SIMULADOR.....	55
FIGURA 25 RESULTADOS DE LA VALIDACIÓN INICIAL CON EL GRUPO DE EXPERTOS.....	58
FIGURA 26 CASO IDEAL.....	60
FIGURA 27 GRÁFICA DE QUEMADO CONSIDERANDO EL ERROR EN LAS PRUEBAS.....	61
FIGURA 28 GRÁFICAS CONSIDERANDO RASGOS DE PERSONALIDAD PARA LA APTITUD DE PROGRAMACIÓN	62

ÍNDICE DE TABLAS

TABLA 1 ENTORNO PRESENTE Y FUTURO DE UN EQUIPO SCRUM	35
TABLA 2 MAPEO DE MSV DE UN EQUIPO DE DESARROLLO DE SCRUM	38
TABLA 3 REPRESENTACIÓN DE LOS ATRIBUTOS DE LOS AGENTES	51
TABLA 4 REPRESENTACIÓN DE LOS ATRIBUTOS DEL MEDIOAMBIENTE	52
TABLA 5 CONFIGURACIÓN DE LA SIMULACIÓN	63
TABLA 6 CONFIGURACIÓN DEL EQUIPO DE DESARROLLO	63
TABLA 7 CONFIGURACION DEL PROYECTO.....	64
TABLA 8 RESULTADOS EXPERIMENTO 2.....	64

Capítulo 1 INTRODUCCIÓN

1.1 Motivación

Una de las primeras tareas para la búsqueda de un modelo de sistema viable para la integración de equipos de trabajo en organizaciones, fue la de definir un caso de estudio, la elección fueron los equipos de desarrollo de software. Esto motivado por la gran complejidad que conlleva el desarrollo de proyectos de software, complejidad que proviene de distintas fuentes entre las que destaca el cliente, quien en ocasiones no es capaz de definir en concreto que es lo que quiere como un producto final, además también tenemos complejidad en el entorno externo ya que los mercados están en constante cambio por lo que las necesidades del cliente en muchas ocasiones cambian en función de los cambios en el exterior y estos cambios deben de ser integrados en cualquier fase del proyecto, esto quiere decir que se requiere de una característica adaptiva en el desarrollo, es importante también mencionar que la complejidad también está presente en el mismo equipo a nivel personal.

Al analizar el enfoque de las metodologías ágiles específicamente con el modelo SCRUM, se encontró una gran similitud en muchas de las ideas presentes en el modelo de Sistemas Viables. Si bien SCRUM no modela a una Organización en su totalidad, su aplicación está pensada a nivel de equipo, Mientras que con Sistemas viables podemos Modelar a una organización entera. Aun que en nuestra investigación nos centramos en los equipos de trabajo en general, para el caso de estudio no se debe dejar de lado a toda la estructura en donde están inmersos los equipos, es decir no es posible analizar a los equipos extrayéndolos de la organización, si se hace esto, además de atentar con la filosofía de viabilidad, estaríamos cayendo en una estrategia reduccionista y limitada al desconectar al equipo de todos los canales de comunicación , colaboración y por todos aquellos canales donde fluyen los recursos para el funcionamiento del equipo. Es por ello que para poder entender el fenómeno de los equipos de trabajo

se requiere de un enfoque holístico que sea capaz de captar la imagen completa del fenómeno.

Con SCRUM podemos ver en práctica mucho de los principios presentes en el enfoque de los Sistemas viables, Entre los que destacan la Auto organización, La comunicación, la colaboración. Esto nos da pauta a analizar si los procesos de la metodología ágil SCRUM se comportan como un Sistema Viable, de ahí que nuestra investigación propone el Modelo de equipos de trabajo con un enfoque de Sistemas Viables.

1.2 Antecedentes

El desarrollo de software es una actividad importante para la sociedad, ya que el software se utiliza en muchos de los productos y servicios que se consumen [1], siendo esta actividad compleja [2]. Es frecuente que los proyectos de software sean realizados por equipos de personas que interactúan entre si y suman esfuerzos con el objetivo de completar con éxito los proyectos que les son asignados. Sin embargo, este escenario puede ser influenciado por diversos factores inherentes y externos al equipo, factores como los requerimientos cambiantes o la presión de la entrega a tiempo (*time-to-Market*) [3] .

Las metodologías ágiles de desarrollo de software surgieron en respuesta a la alta tasa de fallo de las metodologías tradicionales, como el método en cascada [4]. Una de las metodologías ágiles más conocidas es SCRUM, que ha experimentado un crecimiento importante en la industria en el sector de las Tecnologías de la Información (TI) prueba de ello son los reporte anuales acerca del uso de metodologías ágiles como el de SCRUM [5], [6]. SCRUM Básicamente, se trata de un marco de trabajo para el desarrollo de software ágil.

El nacimiento de SCRUM surge en el contexto de empresas japonesas y estadounidenses grandes [4]. Cuando se considera a empresas de este tipo, éstas

frecuentemente cuentan de grandes recursos a su disposición. Sin embargo, en el contexto de las pequeñas y medias empresas (PyMEs), no siempre es así. En algunos casos, es difícil para las PyMEs seguir las recomendaciones de SCRUM por diversos motivos [7], por lo que es posible que estas realicen adaptaciones a SCRUM. Estas adaptaciones pueden ser cambios u omisiones, y un ejemplo de esto es la combinación de SCRUM con otras metodologías (Híbridos de SCRUM) como por ejemplo Scrumban una combinación de SCRUM con Kanban [8].

Dichas adaptaciones pueden ocasionar un des-alineamiento de los ideales de SCRUM al no seguir las recomendaciones. Algunos ejemplos de esto son: (a) las empresas utilizan equipos con un número de integrantes menor o mayor al recomendado, (b) omisión de alguno de los eventos como por ejemplo la retrospectiva, (c) equipos no equilibrados, (d) rotación de personal en los equipos, (e) sprint demasiado largos o cortos, entre otras. En [7], llama anti-patronos (*anti-patterns*) a este tipo de desviaciones y se describe algunos de los más frecuentes.

Aunque se han identificado estos anti-patronos, aún es imprescindible conocer con mayor nivel de detalle los posibles efectos de estos para las empresas, y particularmente en el desempeño de los equipos de desarrollo. Esta es una problemática que necesita ser abordada desde una perspectiva de sistémica y de la complejidad, ya que el desarrollo de software es una actividad socio-técnica compleja [9], realizada por equipos que a su vez son complejos.

Para describir problemas complejos se ha utilizado el modelado computacional que es considerado por algunos autores una tercera forma de hacer ciencia [10]. El Modelado Basado en agentes es una de las formas de modelado computacional [11]. Que consiste en un paradigma científico que permite acercarse a los problemas complejos, mediante la experimentación con la simulación por computadora, esto ayuda a descubrir y describir como es que surgen los comportamientos emergentes que con los métodos tradicionales sería difícil de abordar. Debido a la naturaleza compleja del proceso de desarrollo de

software un modelo basado en agentes nos permite convenientemente describir los procesos de toma de decisiones y comunicación en el sistema.

Algunos trabajos relacionados con herramientas computacionales y SCRUM son:

En [12] se propone una herramienta groupware para reuniones virtuales para proyectos basados en SCRUM. El objetivo de este trabajo es dar soporte a las reuniones especialmente en contextos distribuidos, donde se destaca el uso de la simulación de un ambiente virtual para las reuniones.

Con [13] se propone una herramienta de soporte a los equipos en el proceso de hacer SCRUM, dicha herramienta está basada en web y sistemas multi-agente. Modela el proceso y la conceptualización de SCRUM a través de una ontología.

Otros ejemplos relacionados del uso de SCRUM y los enfoques basados en agentes son las metodologías enfocadas en el desarrollo sistemas multi-agente como en [14].

Los trabajos antes mencionados no son propiamente un modelo de un equipo, sin embargo, utilizan un enfoque de la metodología de Sistemas Multi-agente, dicha metodología está más orientada a la resolución de problemas que al estudio y descripción de un fenómeno, en nuestra propuesta hacemos uso del modelado basado en agentes y el Modelo de Sistemas Viables de forma complementaria para tener un enfoque descriptivo de los equipos de trabajo como los Equipos de en SCRUM.

1.3 Justificación

La necesidad por parte de las organizaciones de poder contar con equipos efectivos que logren los objetivos de manera efectiva y eficiente ha vuelto el área del trabajo en equipo un tópico de interés para la investigación científica.

Nuestra propuesta es un modelo computacional enfocado en el Modelo de Sistemas Viables que aborda aspectos organizacionales en el marco de trabajo de los equipos SCRUM.

La filosofía ágil comparte mucha de las ideas contenidas en el modelo de sistemas viables, por lo que se pretende su integración en el modelo viable para describir el comportamiento los equipos de trabajo en organizaciones de desarrollo de software bajo el marco de trabajo SCRUM.

1.4 Planteamiento del problema

Hoy en día en un entorno social, cultural y tecnológico que tiende cada vez más hacia la colaboración los equipos de trabajo se han convertido en un recurso valioso para las organizaciones, Son muchas las ventajas que tiene el trabajo en equipo, la principal de estas es la sinergia, la cual es un efecto que puede verse cuando los resultados de un equipo son superiores a la suma de los esfuerzos y capacidades individuales de cada uno de los miembros de un equipo. Dicho efecto depende de una buena organización y de que el objetivo sea realmente común, comprendido y aceptado por todos los integrantes del equipo de trabajo, dependiendo en gran medida de la compatibilidad entre los mismos integrantes.

Para las organizaciones el principal interés en cuanto a los equipos de trabajo es que los equipos sean efectivos, es decir, que sean eficaces y eficientes. Para lograr esto las organizaciones utilizan diversos mecanismos como por ejemplo la forma en la que selecciona su personal. Auxiliándose de herramientas provenientes de la ciencia de la conducta como la psicometría, una rama de la psicología utilizada para medir los rasgos de personalidad, el conocimiento, las habilidades, las capacidades mentales o el estado de opinión o actitud.

Estos aspectos han sido utilizados para la selección de personal y la asignación de roles en las organizaciones. Este enfoque se basa principalmente en aplicación de tests o pruebas que arrojan resultados con los que se puede asumir que los individuos tenderán a cierto patrón de conducta o tendrán un desempeño esperado.

Sin embargo, los equipos de trabajo son sistemas complejos ya que su componente básico son las personas, dentro del equipo de trabajo puede ocurrir una amplia variedad de comportamientos emergentes como el conflicto, la cooperación, la colaboración, y la ya mencionada sinergia entre otros.

Aunque las organizaciones pueden disponer de herramientas como las ya mencionadas, actualmente no existen herramientas o modelos que integren todas las características que estas ya miden y describan como es que surge la sinergia sin dejar de lado el aspecto complejo del equipo como las relaciones y las interacciones entre los integrantes y su medio ambiente, además de considerar la estructura de roles, las normas de la organización, etcétera.

Un modelo computacional que integre aspectos como los descritos de una forma conjunta podría ayudar a descubrir nuevas formas en las que se pueda propiciar que el fenómeno emergente y deseado de la sinergia ocurra.

Para esto, se propone abordar a los equipos de trabajo desde la perspectiva de sistemas complejos utilizando el enfoque del Modelo de Sistemas Viables junto con el Modelado Basado en Agentes y la Simulación, para crear un modelo computacional que ayude a complementar los metodos tradicionales y pueda servir de herramienta para describir el comportamiento los equipos de trabajo de una forma aproximada.

1.5 Objetivo General

Creación de un Modelo Computacional de equipos de trabajo en organizaciones con un enfoque de Sistemas Viables.

1.6 Objetivos específicos

- Crear un modelo de Sistema Viable de un equipo
- Crear un Modelo Basado en Agentes del equipo.
- Desarrollar un prototipo de Simulador de equipo
- Validar el modelo

1.7 Estructura de la tesis

La organización de la tesis se describe a continuación: En el primer capítulo contiene la introducción que aborda los temas más relevantes de la investigación, la justificación, planteamiento del problema y los objetivos de la investigación. El capítulo dos contiene el estado del arte en donde se encuentran los enfoques teóricos que sustentan la investigación.

El capítulo tres describe el Modelo de Sistema Viable del Equipo SCRUM, el capítulo cuatro detalla el Modelo Basado en Agentes del Equipo SCRUM, puntualizando en el capítulo cinco el Caso de Estudio donde se genera el prototipo de simulador para equipos de trabajo SCRUM.

Los resultados y diferentes escenarios probados se describen en el capítulo seis para terminar con las conclusiones en el capítulo siete.

Capítulo 2 MARCO TEÓRICO

2.1 Modelo de Sistemas Viables

El Modelo de Sistemas Viables (MSV) fue desarrollado por el cibernético Británico Stafford Beer en su trilogía de libros [15]–[17]. Dicho modelo está inspirado en el sistema nervioso humano y se utiliza como una herramienta para el diagnóstico o el diseño organizaciones [18].

El MSV es un modelo conceptual basado en axiomas, principios y leyes para una organización viable [19]. Desde este enfoque, la viabilidad es la capacidad que tiene un sistema de mantener una existencia separada [20]. Para contar dicha viabilidad la organización debe contener las cinco funciones básicas para la gestión y sus interrelaciones las cuales son: operación, coordinación, control, inteligencia y política [21].

El modelo de sistemas viable denomina a las cinco funciones de viabilidad como sistema 1 al 5 respectivamente. Estos sistemas se interconectan a través de una serie de canales de comunicación por los cuales fluye la información. Estas cinco funciones junto a su enlaces y canales de comunicación constituyen los componentes principales de un MSV. La Figura 1 muestra un diagrama de un Sistema viable.

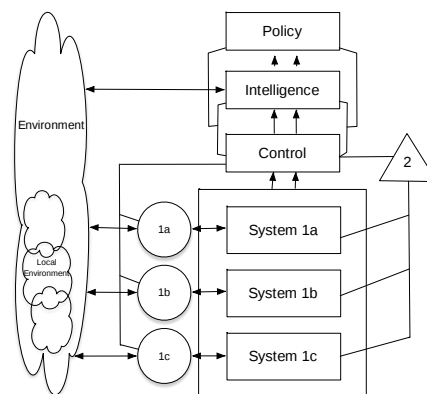


Figura 1 Modelo de Sistema Viable adaptado de [15]

2.1.1 Subsistemas:

2.1.1.1 Sistema 1 Operación:

El sistema 1 es el responsable de producir y entregar al entorno los bienes o servicios que la organización produce. Se refiere a las operaciones fundamentales de la organización, está compuesto por unidades operativas que a su vez son sistemas completos viables, dichas unidades producen lo que se supone la organización debe de producir.

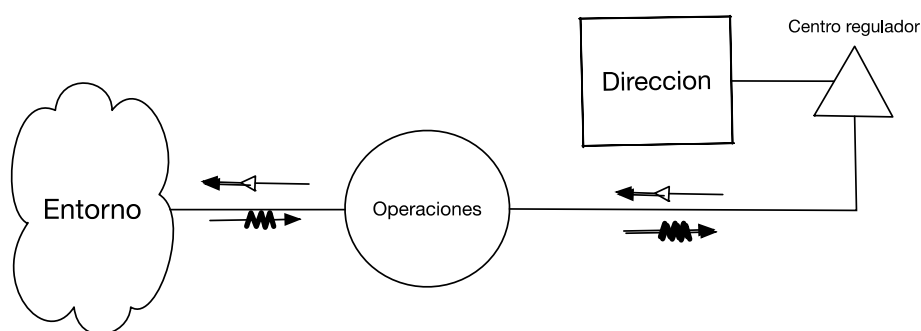


Figura 2 Sistema uno

Cada unidad operativa está integrada por tres elementos: Entorno, Unidad operativa (operaciones) y dirección de la unidad, además de un elemento coordinador (sistema 2) como se muestra en la Figura 2.

2.1.1.2 Sistema 2 Coordinación

El sistema dos se encarga de las actividades de coordinación, su principal función es la de lograr un funcionamiento armónico entre todas las unidades operativas que componen al sistema uno, esto mediante la amortiguación de las oscilaciones que se producen como consecuencia del funcionamiento de dichas unidades operativas y sus interacciones.

Cada unidad operativa (sistema 1) dispone de un sistema dos (local) como se muestra en la Figura 3, todos los sistemas dos locales se conectan al sistema dos

corporativo que se representa con el triángulo trazado con líneas gruesas en la misma figura que se comunica y alimenta al sistema tres con información sobre el funcionamiento de los Sistema uno y en sentido contrario con información para la coordinación de las actividades.

El Sistema dos es un sistema de apoyo para el sistema tres ya que tiene por función absorber la variedad (complejidad) que se genera del funcionamiento de las unidades operativas. Los sistemas dos son los mecanismos de coordinación y resolución de problemas (Los que surgen del funcionamiento e interacción de los sistemas uno) creados para lograr el máximo grado de automatización en el funcionamiento del sistema uno.

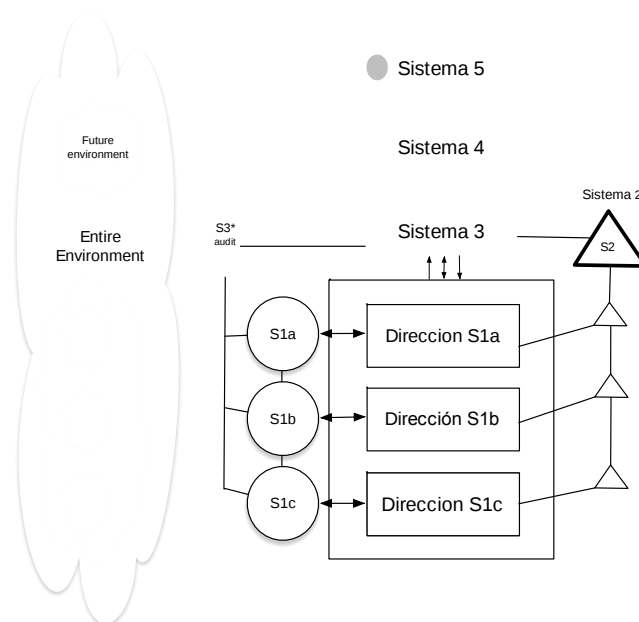


Figura 3 Sistema 2 en color gris

Sistema 3 (Control) (ojo interno)

El sistema tres es el responsable del control interno e inmediato de la organización, por lo cual, este sistema es el encargado de la gestión del conjunto de unidades operativas que constituyen el sistema uno (ver Figura 4), además de supervisar el subsistema dos.

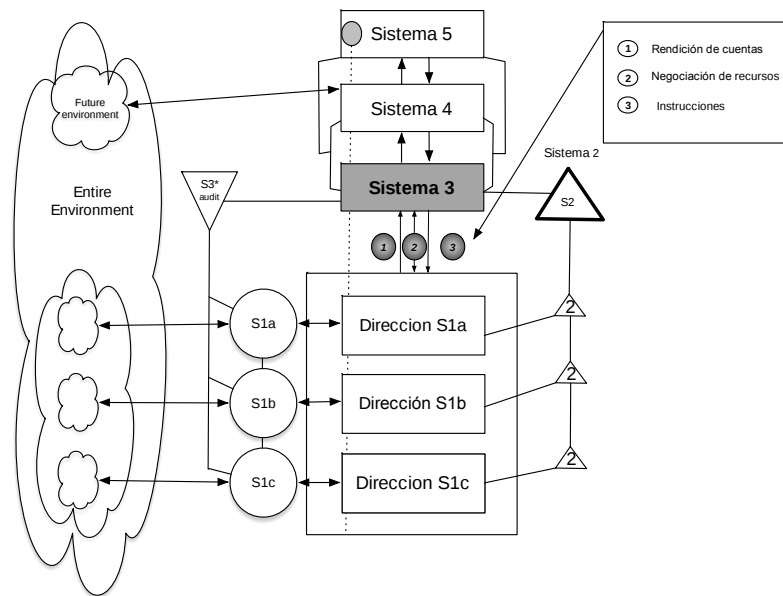


Figura 4 Sistema 3 en color gris

La principal función del sistema tres es asegurar el funcionamiento de la organización en el presente, y logra esto a por medio de:

- Gestionar el conjunto de unidades operativas que conforman el sistema uno.
- Propiciar la integración y el funcionamiento armónico del sistema como colectivo.
- Explotar las sinergias posibles que surgen de la interacción de las unidades operativas que conforman el Sistema uno.
- Ser responsable de lograr la estabilidad interna de la organización.
- Encargarse de asignar los objetivos a cada una de las unidades operativas del sistema uno (en conjunto con el sistema cuatro y en conformidad con el sistema cinco).
- Encargarse de la asignación de recursos a las unidades operativas.
- Ser responsable de establecer los mecanismos de rendición de cuentas de las unidades operativas.

El Sistema tres es también nombrado el “ojo interno” que se encarga del “aquí y del “ahora” de la organización como se muestra en la Figura 5, ya que posee una visión en conjunto de las operaciones que ninguna unidad operativa posee por si sola.

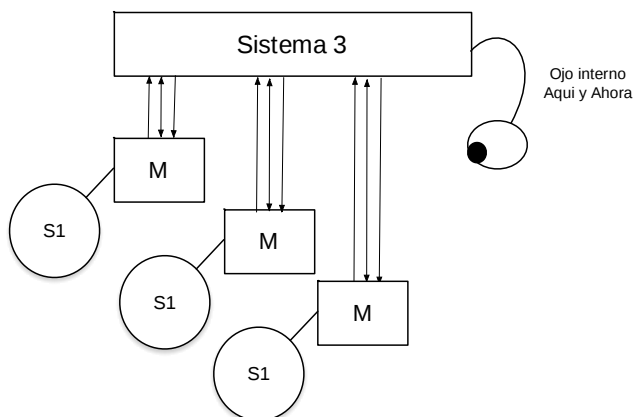


Figura 5 Relación del Sistema tres con cada unidad operativa

2.1.1.3 Sistema 3* Auditoria

El sistema 3* representado con un triángulo invertido color gris en la Figura 6, es un sistema de apoyo al sistema tres, su objetivo es obtener información acerca del funcionamiento del sistema uno que no puede ser obtenida por medio de los canales disponibles al sistema tres (negociación de recursos y rendición de cuentas) ni de la conexión entre el sistema tres y el sistema dos.

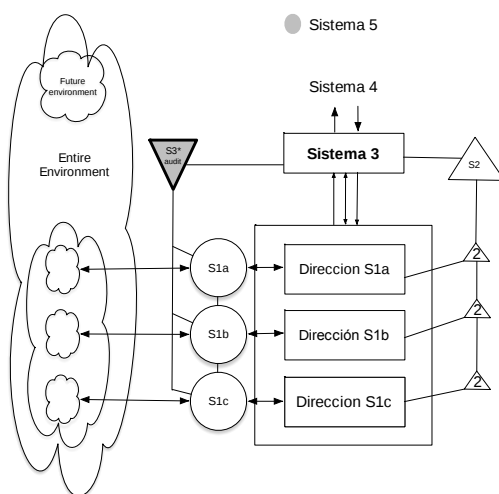


Figura 6 Sistema tres* en color gris

Esta información es complementaria a la información que los Sistema uno y Sistema dos proporcionan al sistema tres. La diferencia entre a información proveniente de los Sistemas 1 y la obtenida la proporcionada por el Sistema tres* es que la segunda es información que puede afectar a la totalidad del sistema 1 (todas las unidades operativas) que no puede ser obtenida por el sistema uno ni el sistema dos.

La función principal del sistema 3* es la de asegurarse que la información que val del sistema uno al tres sea completa. Esta función comúnmente mediante mecanismos como las auditorias esporádicas. Este mecanismo tiene el fin de propiciar que el comportamiento de los componentes del sistema se adecue al deseado.

2.1.1.4 Sistema 4 Inteligencia

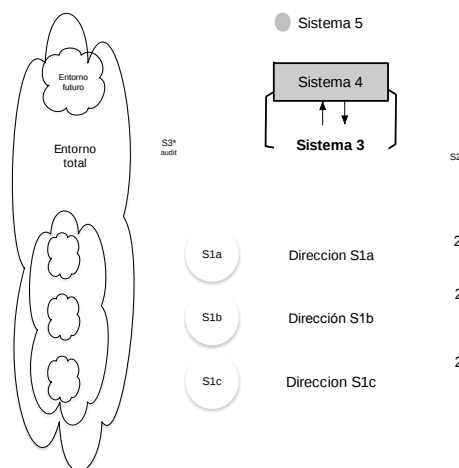


Figura 7 Sistema 4 inteligencia

El Sistema 4 representado en la Figura 7 con un rectángulo en color gris, es el órgano de adaptación de la organización. Su función principal es la de asegurar el funcionamiento de la organización en el futuro además de hacerse cargo del entorno externo de la misma, esto asegurándose de que la organización siga logrando su propósito y mantenga su identidad a pensar de los cambios o

amenazas que puedan surgir en su entorno externo. Debe generar la imagen de la propia organización y mostrarla en el entorno externo.

La dirección del Sistema 4 es de carácter estratégico, para lograr su función el Sistema 4 debe disponer de herramientas como los sistemas de monitorización, que provean información tanto del presente como del futuro (posibles cambios) con dicha información el Sistema 4 debe ser capaz de anticiparse a dichos cambios y realizar los cambios a tiempo que sean necesarios para mantener su viabilidad. Algunas de estas herramientas para el apoyo en la toma de decisiones estratégicas son: los estudios de perspectiva, Dinámica de Sistemas.

2.1.1.5 Sistema 5 Política

El Sistema 5 representado con un rectángulo gris en la Figura 8, es la autoridad máxima dentro de la organización, ejerce el tipo de dirección normativa. Por lo cual es el único con la capacidad para regular la interacción que existe entre el sistema 3 y el sistema 4. Todo la variedad (problemas) que dichos sistemas no son capaces de absorber (resolver) debe ser absorbida (resuelta) por el Sistema 5 en su función de cierre del sistema es decir la última autoridad de la organización.

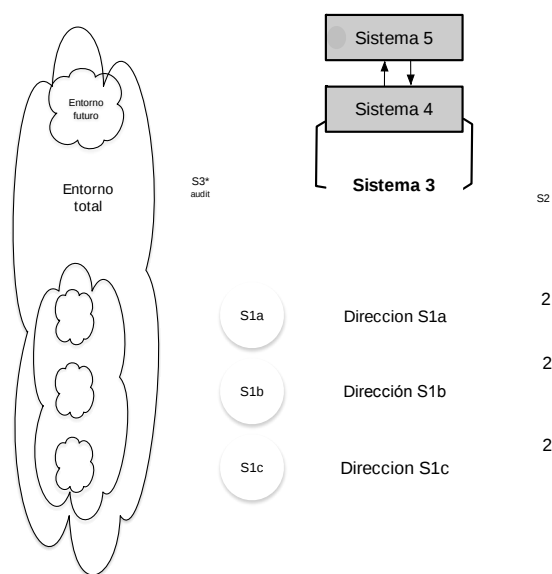


Figura 8 Sistema 5 Política

La función reguladora del Sistema 5 consiste en equilibrar el presente (Sistema 3) y el futuro (Sistema 4) de la organización, para esto el Sistema 5 debe tener en cuenta los aspectos tanto internos como externos.

Este Sistema es el responsable de definir la identidad de la organización. La identidad de la organización consiste en definir qué es lo que quiere ser la organización y también que no quiere ser [22].

Las principales funciones del Sistema 5 son:

- Suministrar el cierre lógico del subsistema viable
- Regular el homeostato Sistema 3 – Sistema 4
- Establecer con claridad los límites de la organización (que forma parte del entorno y que es lo que forma la organización).
- Establecimiento de las normas y las reglas de conducta que rige a todos los niveles de recursión.
- Establecimiento de la visión, misión y objetivos estratégicos.
- Asegurarse de mantener la identidad de la organización.

2.2 Modelado Basado en Agentes y Simulación

Los Modelos Basados en Agentes y Simulación (ABM por sus siglas en ingles) son modelos computacionales que se utilizan como una de las herramientas para modelar sistemas complejos [23] [24]. Esta forma de modelado es un enfoque relativamente nuevo para el modelado los sistemas complejos. Los componentes básicos de un ABMS que se muestran en la Figura 9 son: a) los agentes, b) su medio ambiente y c) los mecanismos de interacción entre los agentes [24].

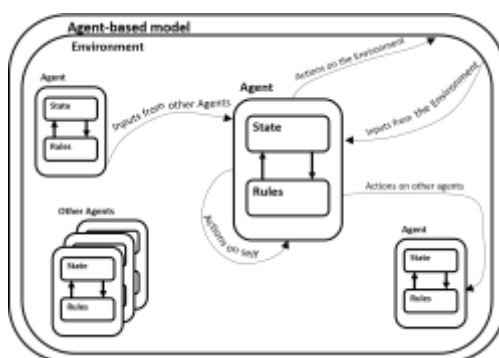


Figura 9 Esquema general de un ABMS adaptado de (Sayama, 2015)

El bloque básico de modelado en ABMS son los agentes [25]. Aunque no existe un consenso sobre una definición universal de un agente, las definiciones tienden a coincidir en más puntos de los que no están de acuerdo [26], en este artículo seguimos la siguiente definición de un agente::

“Los agentes son programas de computadora, o distintas partes de un programa que son utilizados para representar actores sociales: personas, organizaciones como empresas, u organismos como naciones”. Estos agentes son programados para reaccionar al entorno computacional en donde están localizados. Donde su medio ambiente es un modelo del ambiente real donde el actor social actúa” [27].

Los agentes tienen la capacidad de exhibir comportamientos y tomar decisiones autónomas, todo ello regido por reglas simples. Dichas reglas de comportamiento

son descritas de una forma algorítmica en lugar de una forma puramente matemática [24].

En [28] se enlistan algunas características esenciales de los agentes basadas en aplicaciones recientes, estas son: a) un agente es auto-contenido, b) un agente es autónomo y auto-dirigido, c) un agente tiene un estado que varía en el tiempo, d) un agente es social teniendo interacciones dinámicas con otros agentes que influyen su comportamiento.

Otras características también mencionadas en [28] son a) un agente puede ser adaptable, b) un agente puede ser orientado a metas, c) un agente puede ser heterogéneo.

Los modelos basados en agentes usualmente son implementados como una simulación por computadora, dicha simulación puede servir como un remplazo virtual del campo de experimentación para los investigadores [29][24].

En nuestro caso de estudio, se utilizó este enfoque como herramienta de modelado computacional considerando las características heterogéneas de los integrantes de un equipo de trabajo, las interacciones no lineales entre sus miembros y su medio ambiente.

2.3 Metodología Ágil SCRUM

SCRUM no es una metodología en el sentido estricto de la palabra, sin embargo, diversos autores la catalogan como una metodología con fines prácticos. SCRUM es la metodología más utilizada dentro de los métodos ágiles según reportes como [5][6].

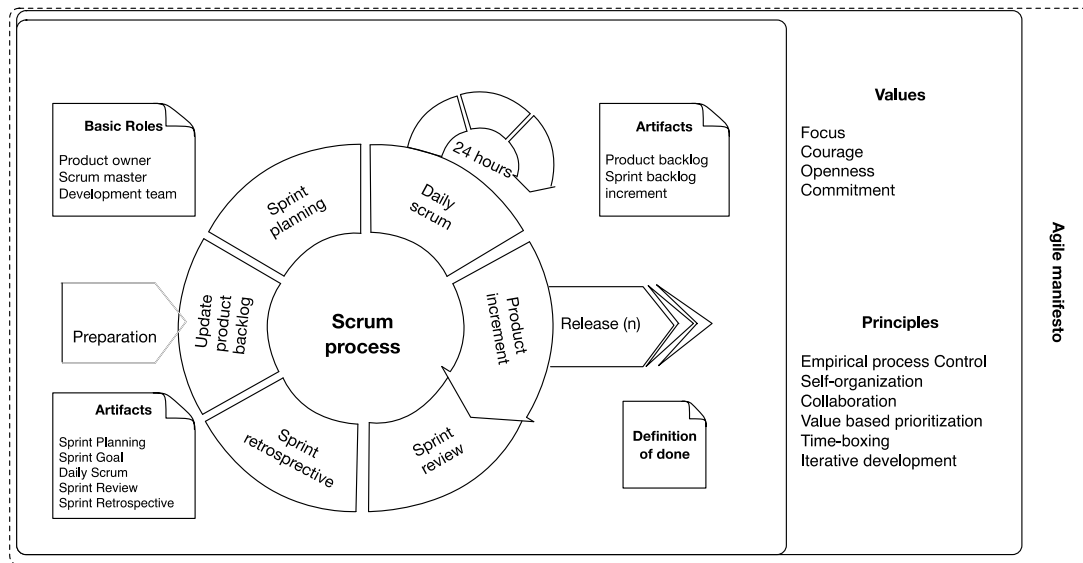


Figura 10 Marco de trabajo SCRUM

SCRUM es “Es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos productivamente y creativamente con el máximo valor”[31].

Dicho marco utiliza un enfoque iterativo e incremental y el control de procesos empíricos [32]. Esta es la metodología ágil de mayor adopción de acuerdo a varios reportes [33]. En la Figura 10 se muestra el proceso los componentes y aspectos de SCRUM que se describen a continuación.

La definición de SCRUM consiste en los roles, eventos y artefactos y reglas asociadas; cada uno de dichos componentes sirve a un propósito específico y es esencial para el éxito de SCRUM y para su uso [31]. SCRUM define tres roles básicos: Dueño del producto (PO por sus siglas en ingles), (2) Maestro SCRUM (SM por sus siglas en inglés) y (3) Equipo de Desarrollo (DT por sus siglas en ingles); seis eventos (1) El sprint (2) Planeación del sprint (3) SCRUM diario (4) Revisión del sprint (5) Sprint y (6) retrospectiva; y tres artefactos: (1) Pila del producto (PO por sus siglas en ingles) (2) Pila del sprint (SB por sus siglas en inglés) y (3) el incremento.

Tiene tres pilares que soportan la implementación del control de procesos empírico: 1) transparencia, 2) Inspección y 3) Adaptación. La transparencia se refiere a que todos los aspectos significativos para el proceso deben de ser

visibles a todos aquellos que son responsables del resultado, esto requiere que los aspectos sean definidos con un estándar común para que todos los involucrados compartan un entendimiento común acerca de la información. La inspección se refiere a la inspección frecuente de los artefactos de SCRUM con el propósito de detectar desviaciones innecesarias que puedan resultar en un producto inaceptable. La adaptación se refiere, a los ajustes que pueden realizarse si las desviaciones detectadas se encuentran fuera de los límites permisibles con el fin de minimizar desviaciones mayores.

SCRUM define cinco valores que rigen toda la implementación del marco, dichos valores están en línea con el manifiesto ágil y son: 1) Compromiso, 2) Coraje, 3) Foco, 4) Apertura y 5) Respeto.

2.3.1 Roles de un Equipo SCRUM

El sistema de roles que define SCRUM que se muestra en la Figura 11 consiste en: (a) el Dueño del Producto (*Product Owner*), quien es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo, (b) el equipo de Desarrollo (*Development Team*) ó el Miembro de Equipo (*Team member*), que forma parte del equipo de desarrollo, (c) y finalmente el Maestro SCRUM (SCRUM Master), quien desempeña el papel del líder del equipo y además es responsable de velar por el equipo y asegurarse del entendimiento y la práctica correcta de SCRUM [31].

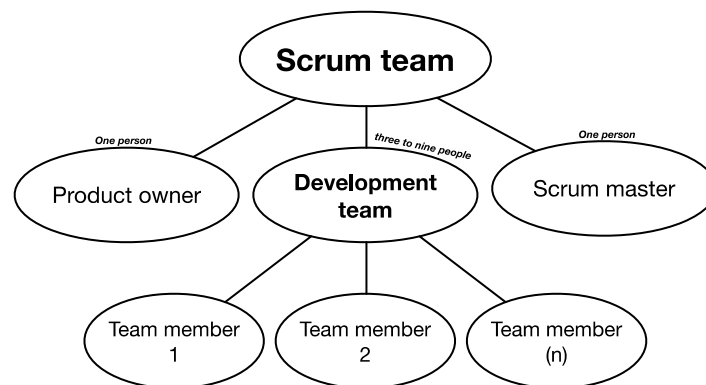


Figura 11 Composición de un Equipo SCRUM

2.3.1.1 *Dueño del producto (PO)*

El dueño del producto es el responsable de maximizar el valor del producto y el trabajo del DT [31]. El PO es la única persona responsable de gestionar el PB además de ser la única persona que tiene la facultada de cancelar un sprint; sin embargo, él puede hacerlo estando bajo la influencia de los interesados o los demás miembros del ST.

2.3.1.2 *Equipo de Desarrollo (DT) o Miembro del Equipo (TM)*

Está formado por todos los individuos necesarios para la construcción del producto. El Miembro del Equipo es el responsable de la construcción y calidad del producto. El ED es auto-organizado. Esto significa que no existe un líder externo que asigne las tareas ni que determine la forma en que serán resueltos los problemas. Es el mismo equipo quien determina la forma en que se realizara el trabajo y como resolverá la problemática que se presente.

Cada ME debe poseer todas las habilidades necesarias para poder realizar el trabajo requerido. Esta característica se conoce como multifuncionalidad y significa que dentro del ED no existen especialistas exclusivos. Si no más bien individuos generalistas con capacidades especiales. Lo que se espera de un ME de desarrollo es que no solo realice las tareas en las cuales se especializa sino también en todo lo que este a su alcance para colaborar con el éxito del equipo.

Las principales responsabilidades del ME son:

- Proveer las estimaciones de cuanto esfuerzo será requerido para cada una de las características del producto.
- Comprometerse al comienzo de cada sprint para construir un conjunto determinado de características en el tiempo que dura el mismo sprint.
- Responsable por la entrega del producto terminando al finalizar cada sprint

2.3.1.3 *Maestro SCRUM (SM)*

El SM es el responsable de asegurar que el equipo de SCRUM trabaje ajustándose a la teoría, prácticas y reglas de SCRUM. Es quien vela por la utilización de SCRUM, la remoción de impedimentos y asiste al equipo a que logre su mayor performance posible. Es considerado un coach o facilitador encargado de acompañar al ED [31].

Principales Responsabilidades:

- Velar por el correcto empleo y evolución de SCRUM
- Facilitar el uso de SCRUM a medida que avanza el tiempo
- Asegurar que el equipo de desarrollo sea multifuncional y eficiente
- Proteger al equipo de desarrollo de distracciones y trabas externas al proyecto
- Detectar, monitorear y facilitar la remoción de los impedimentos que puedan surgir con respecto al proyecto y a la metodología
- Asegurar la comunicación dentro del equipo
- Detectar problemas y conflictos interpersonales dentro del equipo de trabajo (para respetar la filosofía auto organizativa lo ideal es que el equipo mismo sea quien resuelva estas cuestiones) en caso de que no sea posible se deberá involucrar al SCRUM master y eventualmente a niveles más altos de la gerencia.

2.3.2 *Eventos*

Los eventos de SCRUM son bloques de tiempo (time-boxes) que se desarrollan durante cada iteración de SCRUM, a continuación, se describen cada uno de ellos:

2.3.2.1 El Sprint

El sprint es el corazón de SCRUM. Este es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento del producto utilizable y potencialmente disponible. Un sprint puede verse como un proyecto entero, que contiene todo lo necesario para serlo, es decir, este tiene un propósito, una meta y plan propio.

La meta del sprint se establece antes del inicio del sprint y esta se alcanza cuando se completan todos los elementos de la pila del producto seleccionados para pertenecer a un sprint como se muestra en la Figura 12. Esta meta sirve como enfoque para el DT ya que ayuda al equipo a entender porque se construye el incremento.

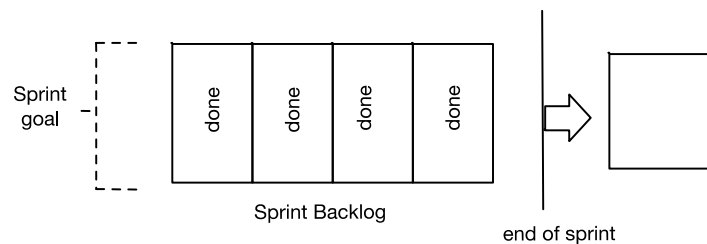


Figura 12 Meta del sprint

Un Sprint está compuesto por: a) reuniones de planificación del Sprint (*sprint planning meeting*), b) los SCRUMS diarios (*daily SCRUMs*), c) el trabajo de desarrollo, d) la revisión del Sprint (*sprint review*), e) y la retrospectiva del Sprint (*sprint retrospective*) [31].

Durante la ejecución de un Sprint no es posible realizar cambios que puedan afectar en la meta del sprint, los objetivos de calidad no disminuyen y el alcance del mismo puede clarificarse y renegociarse entre el dueño del producto y el equipo de desarrollo.

2.3.2.2 Planeación del Sprint (Sprint Planning)

La Planeación del Sprint (SP) es una reunión en donde se planifica el trabajo a realizar durante el sprint. Dicha planificación se crea mediante el trabajo colaborativo del ST completo, aunque el DT puede invitar a otras personas con la finalidad de obtener asesoría técnica o de dominio. La caja de tiempo (time-box) para esta reunión es de máximo ocho horas para un sprint de un mes, para Sprint más cortos suele ser más corto. Durante esta reunión se responden dos preguntas: 1) ¿Que se obtendrá en el Incremento resultado del Sprint? ¿y 2) Como se pueden cumplir las tareas necesarias para la entrega del Incremento a realizar?

Al término de esta reunión el DT deberá haber definido una meta para el sprint, una pila del sprint, y una asignación inicial de tareas. Para todo esto, se deberá considerar como entradas la pila del producto priorizada, el último incremento, las mejoras e impedimentos de la retroalimentación, la capacidad y el desempeño del DT, como se muestra en la Figura 13.

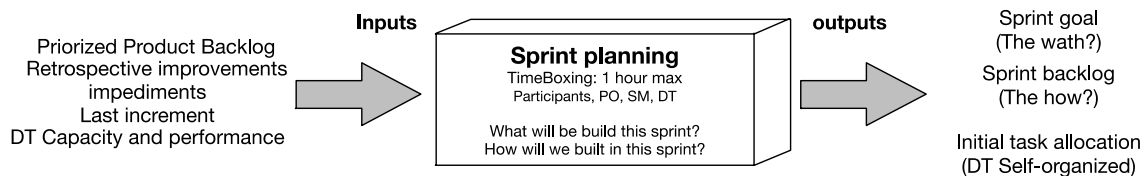


Figura 13 Entradas y salidas de la planeación del sprint

2.3.2.3 SCRUM Diario (Daily SCRUM)

El SCRUM diario es un evento de SCRUM, que consiste en una reunión con una bloque de tiempo (Time-box) de 15 minutos llevada a cabo siempre en el mismo lugar. Tiene el propósito de que el DT sincronice sus actividades y cree un plan para las siguientes 24 horas. Este evento ayuda en los aspectos de inspección y adaptación ya que en él se pueden identificar impedimentos y mejoras. El DT es responsable de dirigir dicho evento, mientras que el SM se asegura se realice y se respete el tiempo establecido.

2.3.2.4 Retrospectiva del Sprint (Sprint Retrospective)

En el desarrollo ágil, las retrospectivas están recomendadas para ser aplicadas continuamente, como es el caso de SCRUM [34]. Este evento de SCRUM, tiene lugar después de la revisión del sprint y antes de la siguiente planeación del sprint. Dicho evento consiste en una reunión con un time box de tres horas para sprint de un mes y para sprint más cortos se reserva un tiempo proporcionalmente menor. Este evento tiene el propósito de inspeccionar como fue el último sprint, enfocándose en la inspección y adaptación, es decir es un momento en donde se pueden identificar mejoras podrían implementarse ya sea en el siguiente sprint o posteriormente.

2.3.3 Artefactos

Los artefactos de SCRUM son herramientas que tienen la finalidad de representar el trabajo o valor, estos son: (a) la lista del producto (*product backlog*), consiste en una lista ordenada de todo lo necesario en el producto; (b) el seguimiento del progreso hacia un objetivo: algunas de las prácticas de proyección para predecir el progreso son las gráficas de trabajo consumido (*burn down charts*), trabajo avanzado (*burn up chart*) y flujo acumulado (*cumulative flow*); (c) la lista de pendientes del Sprint (*sprint backlog*) que contiene el conjunto de elementos de la lista de producto seleccionados para el Sprint y además un plan para entregar el incremento de producto y conseguir el objetivo del Sprint [31]. Para conocer más acerca de SCRUM se le sugiere revisar la siguiente guía de SCRUM [31].

2.3.3.1 Pila del producto (Product Backlog)

El PBL es una lista ordenada de todos los requerimientos necesarios para el producto este puede contener: Características, funcionalidades, requisitos, mejoras y correcciones. Inicialmente la lista refleja los requisitos conocidos y mejor

entendidos al principio. El PB es un artefacto dinámico y vivo ya que evoluciona a medida que el producto y el entorno en el que se usara también lo hacen. Esta lista es gestionada exclusivamente por el PO ver la Figura 14.

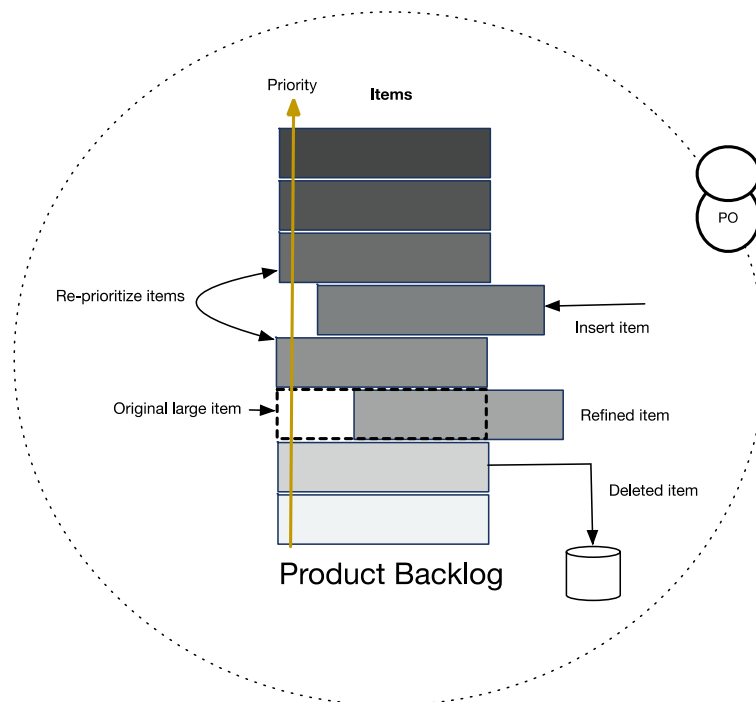


Figura 14 Pila del producto

2.3.3.2 Pila del sprint

La pila del sprint es el conjunto de elementos de la pila del producto seleccionados para el sprint, más un plan para entregar el incremento de producto y conseguir el objetivo del sprint. Solo el equipo de desarrollo puede cambiar la pila del sprint durante la ejecución del sprint. Este artefacto puede ser visto como una imagen visible en tiempo real del trabajo que el equipo de desarrollo planea llevar a cabo durante el sprint [31].

2.3.3.3 Incremento

El Incremento es la suma de todos los elementos completados de la pila del producto durante un sprint, más el valor de los incrementos de todos los sprint previos. Un incremento debe estar en condición de ser utilizado si el PO lo decide liberar a producción [31].

Capítulo 3 Modelo de Sistema Viable de un Equipo SCRUM

En este trabajo de investigación tomamos como caso de estudio la estructura de una organización dedicada al desarrollo de software. La investigación estudia la aplicación del modelo de sistemas viables (MSV) a una organización de este tipo, donde proponemos un modelo que identifica el lugar que ocupa cada elemento de la organización de la empresa de desarrollo de software en el MSV, es decir, se realiza un mapeo de los elementos de la organización en el MSV.

El modelo que se propone es un modelo de cuatro niveles de recursión: proyecto, equipo SCRUM, equipo de desarrollo y miembro de equipo, en este trabajo en particular el sistema focal es el equipo de desarrollo, por lo que el MSV aquí propuesto es el del equipo. para llegar al modelo se siguió la metodología de [35], A continuación, se describen los pasos que definen nuestro MSV de un equipo de desarrollo de software en SCRUM.

3.1 Identidad del sistema

Aunque el uso del marco de SCRUM no es exclusivo de organizaciones de TI, nosotros nos centramos en su aplicación nicho: El desarrollo de software. Actualmente, con el auge del paradigma ágil y de SCRUM como el principal exponente, Organizaciones diversas de diferente tamaño y naturaleza las cuales desarrollan software ya sea como producto o para uso propio han adoptado SCRUM. Son dichas organizaciones las que son de interés para esta investigación. Como se mencionó existe una gran diversidad de organizaciones que utilizan SCRUM por lo que el enfoque o estrategia que se sigue aquí es la siguiente: sea cual sea la estructura o naturaleza de la organización, si esta desarrolla software y utiliza SCRUM entonces deberá existir un departamento o área quienes finalmente serán los responsables del desarrollo de los proyectos que implican el desarrollo de software, a excepción de las organizaciones que contratan a otras empresas para cubrir esta necesidad. sin embargo, en los casos

en los que se utiliza el marco SCRUM existirán los equipos SCRUM. Dichos equipos son nuestro objeto de estudio, en el cual se realiza el análisis diagnóstico a través del MSV y estos se abordan desde una estructura organizacional genérica como se mostrará en la **¡Error! No se encuentra el origen de la referencia..**

El propósito de un equipo SCRUM es: desarrollar proyectos Software siguiendo el marco de SCRUM. Un Equipo de SCRUM está embebido dentro de una organización la cual forma parte de su entorno, además, dicho entorno incluye todo aquello relacionado con la ejecución del proyecto: Comienza de una necesidad (un nuevo software o el mantenimiento o mejora de uno existente) y termina con la entrega de un producto que satisface las necesidades del cliente. En la Figura 15 se muestra a un equipo SCRUM como una organización que interactúa con su entorno.

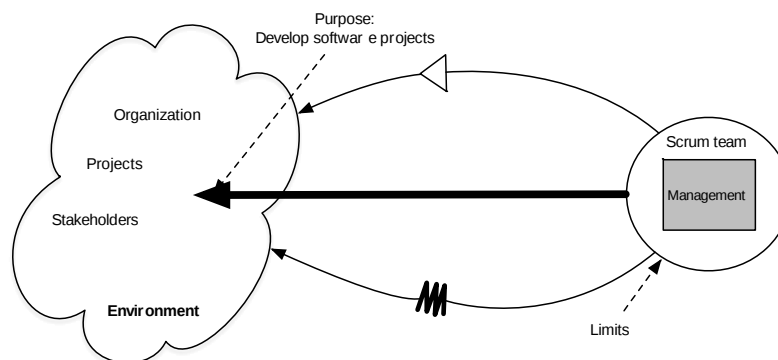


Figura 15 Propósito y límites de un equipo SCRUM

Todos los elementos intermedios entre el comienzo y la finalización del proyecto son parte del entorno de un equipo SCRUM. Por lo tanto, Los límites de un equipo definen lo que es, lo que no debe ser y lo que debe hacer, esto está dentro del marco de SCRUM es decir en sus componentes que son tres cosas: (1) la definición de roles con sus funciones y responsabilidades, (2) la definición un conjunto de eventos, cada uno de ellos con un propósito y reglas definidas y por último (3) un conjunto de artefactos como fuente de información relevante y

mecanismos de coordinación para el equipo. El límite, el propósito y el medio ambiente de un equipo SCRUM se ilustran en la Figura 15.

3.1.1 Entorno presente y futuro

El entorno de un equipo SCRUM está dado por la organización en la cual está embebido, el marco de SCRUM y el proyecto asignado. Estos entornos pueden estar relacionados con dos dimensiones: entorno futuro y entorno presente. En la Tabla 1 especificamos los entornos más relevantes para el ST que nosotros identificamos.

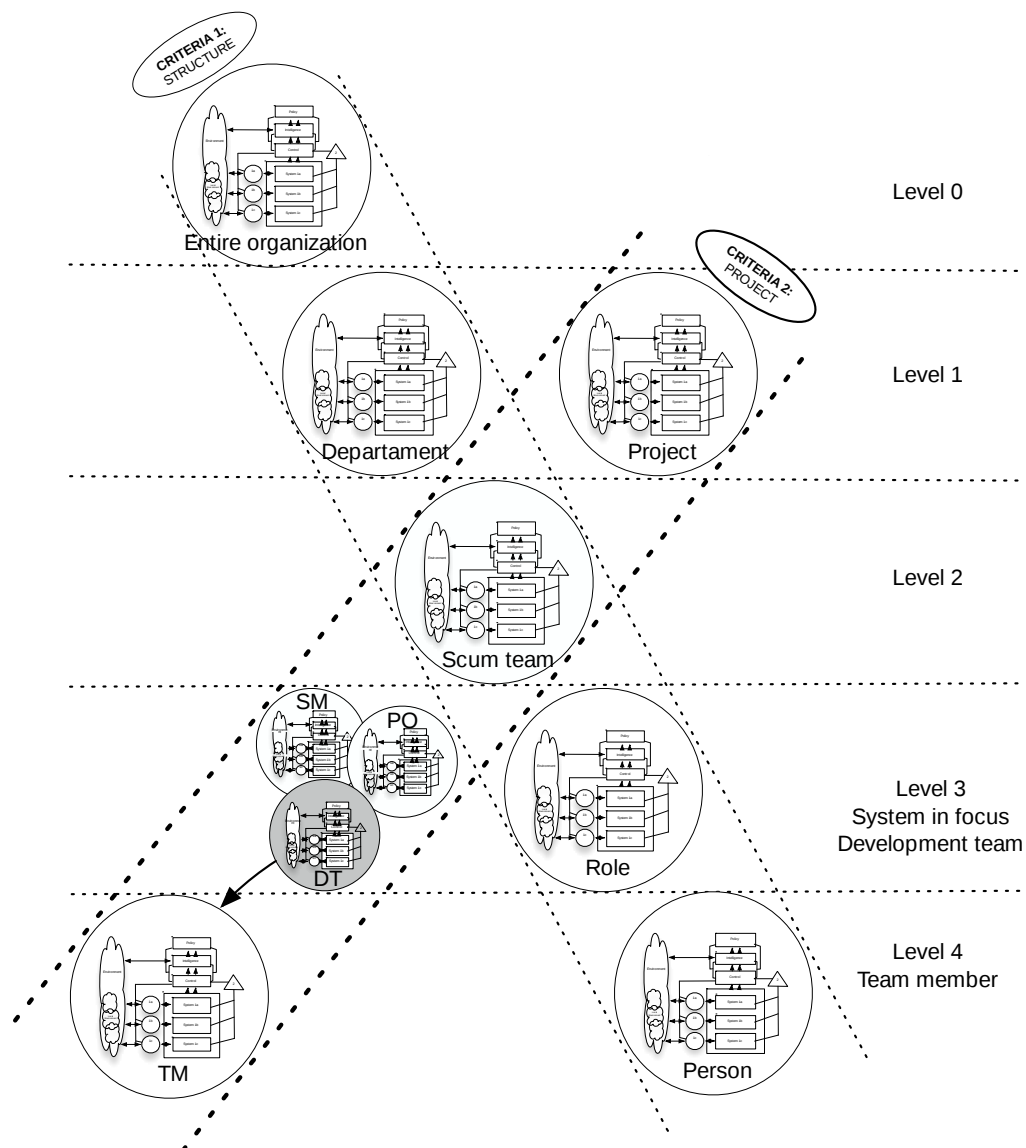
Tabla 1 Entorno presente y futuro de un equipo SCRUM

Entorno presente	Entorno futuro
• Sprint actual • Pila del sprint • Propósito del sprint • Tareas en curso • Recursos actuales • Impedimentos actuales • Capacidad del equipo • Forma de auto-organización • Información actual de los artefactos • Eventos en curso • Definición de hecho • Valores de SCRUM • Manifiesto ágil	• Visión del proyecto • Planeación del siguiente sprint • Meta del sprint • Impedimentos futuros • Cambios en el proyecto • Pila del producto • Eventos de SCRUM • Estimaciones • Mejoras por implementar • Capacitación

3.2 Dimensión vertical

3.2.1 Desdoblamiento de la complejidad: criterios y niveles de recursión

En la Figura 16 se muestra la esfera de recursion que nosotros identificamos, en la figura se muestran dos criterios de recursion: Estructura organizacional y Proyecto, esta ultima es la que utilizamos para abordar a los equipos de desarrollo



de SCRUM, esta consiste en cuatro niveles de recursión: Proyecto, Equipo Scrum, Equipo de desarrollo y Miembro de equipo.

3.3 Dimensión horizontal

3.3.1 Sistema focal: El equipo de desarrollo

Nosotros definimos como el sistema focal el DT que puede verse en la Figura 16 en un círculo relleno de color gris sólido, nosotros definimos dicho sistema como focal ya que es a este nivel donde podemos analizar la influencia del SCRUM Framework que comienza en nivel 2 (ST), y un nivel abajo los aspectos individuales (nivel 4). Además, otra razón de nuestra elección es, que el DT está formado por los profesionales (TMs) que, en último término, son los responsables de desarrollar el proyecto. En la figura 17 se muestra el MSV de un DT, Como se puede apreciar este compuesto por los TMs.

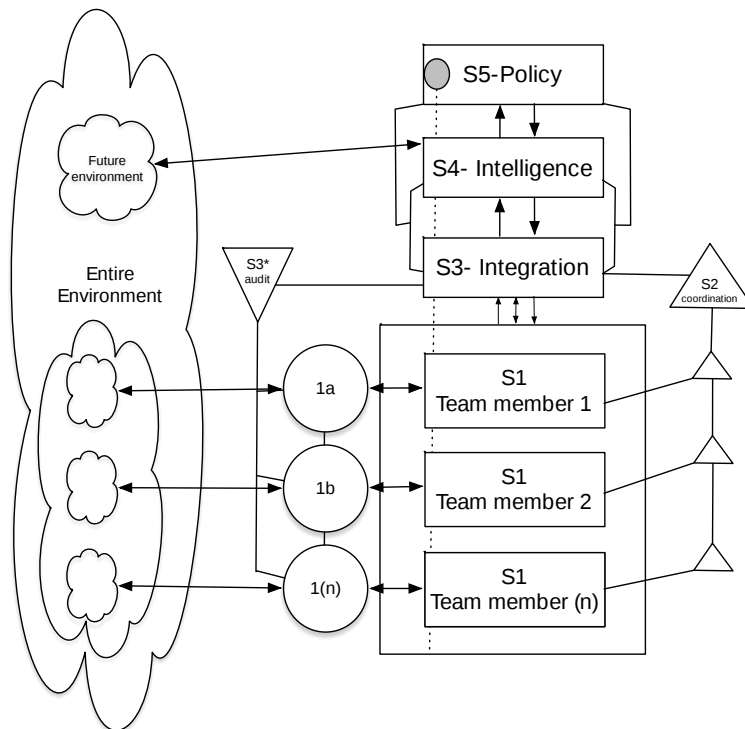


Figura 17 Modelo de Sistema Viable de un Equipo de desarrollo SCRUM

3.3.2 El sistema completo y sus funciones de viabilidad

En este punto se realiza analizan cada uno de los sistemas que componen al sistema en foco, en la Tabla 2 se muestra un mapeo de SCRUM y su función de viabilidad para el equipo de desarrollo SCRUM. Posteriormente se aborda cada a detalle.

Componente		S1	S2	S3	S3*	S4	S5
Roles	Product Owner			●		●	●
	SCRUM Master		●	●	●	●	●
	Team member	●					
Eventos	Sprint		●				
	Sprint planning			●			
	Sprint goal						●
	Daily SCRUM		●				
	Sprint review			●		●	
Artefactos	Sprint retrospective		●	●	●	●	
	Product backlog					●	
	Sprint backlog			●			
	Increment*						
Aspectos	Monitoring charts			●			
	Definition of done						●
	SCRUM Vision and Theory						●
	Agile manifesto						●
	SCRUM values						●
	SCRUM principles						●

Tabla 2 Mapeo de MSV de un Equipo de desarrollo de SCRUM

3.3.2.1 Sistema 5 Política

Las funciones del S5 están distribuidas en varios aspectos que provienen del SCRUM Framework, los elementos que identificamos para el sistema en foco son: el manifiesto ágil, los valores de SCRUM, la guía oficial del SCRUM, la visión del producto, la definición de hecho y algunas funciones del DP y MS.

- **S5 - Manifiesto ágil**

El Agile-manifestó descrito anteriormente, tiene una función de S5 ya que rige a toda la implementación de SCRUM, por lo que está presente en todos los niveles de recursión donde están presente los componentes de

SCRUM incluyendo el sistema en foco. Este documento es la esencia de la definición de agilidad y es un pilar fundamental de la constitución de la identidad de un ST y sus componentes (por ejemplo, el DT). Este documento puede verse como un estándar común de todos los practicadores de las metodologías del desarrollo ágil que incluye SCRUM.

- **S5 Valores de SCRUM**

Los valores de SCRUM al igual que el Agile manifestó tienen una función de S5 ya que son un estándar común. Estos valores son la base para los procesos e interacciones del equipo, la adopción de dichos valores tiene el propósito de propiciar la salud del equipo y el alcance del éxito.

- **S5 – Guía oficial de SCRUM**

Otro documento que también desempeñan una función de S5 es la Guía oficial de SCRUM[31]. Esta guía fue desarrollada por los creadores de SCRUM, es la referencia principal sobre SCRUM. En ella se encuentra la definición Formal de SCRUM más aceptada. y la base de otra literatura. Esta guía desempeña una función de tipo S5 ya que define como su portada dice esta guía contiene “las reglas del juego” y toda la implementación se rige bajo dicha definición.

- **S5 - Definición de hecho**

Este artefacto tiene una función de tipo S5, ya que es una definición que es compartida por todos los equipos SCRUM y que el equipo de desarrollo utiliza para informar que se ha completado un elemento o un incremento. Este común acuerdo puede verse como una política conocida y entendida en todo el ST.

- **S5 - Maestro**

El maestro SCRUM, es uno de los roles definidos en SCRUM, este tiene la función S5, ya que es la figura principal que asegura que todos los

involucrados (incluido el DT) entiendan SCRUM y lo apliquen de la mejor manera. El SM es un experto de SCRUM, por lo que puede ser visto como una entidad que tiene el conocimiento de un experto en SCRUM, Su función no se limita solo a ser un consultor de SCRUM, sino que usa y aplica ese conocimiento para asegurar una implementación correcta de SCRUM, ejemplos de esto son: asegurarse que todos los eventos sean realizados, los bloques de tiempo sean respetados, y haya desviaciones de los propósitos de cada una de estas reuniones.

- **S5- Propietario del producto**

El DP es responsable del éxito del producto y su retorno de la inversión (ROI por sus siglas en ingles) [36]. Esta figura desempeña funciones dentro del S5, ya que es el DP quien representa al cliente, él es el experto que conoce todo lo relacionado con el negocio (producto) y se encarga de definir las prioridades en función de las necesidades actuales del cliente. Por lo que nadie tiene el poder de decirle al DP que trabaje con un conjunto diferente de requisitos. El DP es quien aprueba los incrementos. Para que un DP tenga éxito, toda la organización debe respetar sus decisiones. El DP es quien dirige la creación de la visión del proyecto y además toma las decisiones ejecutivas del más alto nivel en la esfera de recursión definida colocando dentro del S5.

- **Visión del proyecto**

Es la esencia de lo del proyecto o producto que el equipo SCRUM y el equipo de desarrollo están intentando crear. Los Equipos SCRUM pueden crear una visión general del producto y ser descompuesta en un conjunto levemente más detallado por cada incremento, ya que cada Sprint es un mini proyecto completo en sí mismo. La forma en que el Equipo documenta dicha visión depende de cada equipo, un ejemplo puede ser una pequeña presentación de diapositivas.

3.3.2.2 Sistema 4 Inteligencia

Las funciones del S4 se encuentran distribuidas en varias entidades del Sistema en foco, los elementos que identificamos para el sistema en foco son: Dueño del producto, Pila del producto.

S4- Maestro SCRUM

- El objetivo del MS es el promover que el DT trabaje de la mejor forma posible en línea con el marco de SCRUM. El MS desempeña funciones dentro del S4, estas son aquellas que están relacionadas con su rol de líder-facilitador, por ejemplo, cuando el MS se encarga de anticiparse con algo que pudiera interferir con el trabajo del equipo de desarrollo además de los impedimentos esta es una función dentro del S4.

- **S4- Dueño del producto y Pila del producto**

La pila del producto es una especie de modelo compartido (creado por el DP) de lo que el DT debe lograr para satisfacer al cliente, este consiste en una lista de elementos (historias de usuario) ordenadas por prioridades, es decir, la lista comienza con de los elementos de más alta prioridad (para el negocio) hasta los menos prioritarios. Esta lista no es estática por el contrario es un artefacto vivo por lo que puede modificarse según las nuevas necesidades del cliente (cambio de prioridad o la agregación de nuevos elementos) esto únicamente por el PO. Esto coloca al PB junto con la pila del producto como parte del S4 ya que la pila del producto es el ambiente futuro (contiene los elementos siguientes a desarrollar) y el DP se encarga de gestionar dicho artefacto de acuerdo a las necesidades del cliente que incluye el aspecto futuro, para lograr una gestión adecuada del artefacto el DP deberá estar en constante comunicación con las partes interesadas (clientes).

3.3.2.3 Sistema 3 Integración

El S3 es un elemento integrador, que se encarga de la gestión del conjunto de unidades operativas (S1) [37]. Este sistema debe asegurarse de garantizar que dicho conjunto funcione en línea del interés del todo (global). Esto podría no resultar ventajoso para una o más partes en un momento dado, de modo que existen decisiones operativas de las que el S3 debe encargarse, para realizar su función el S3 dispone de tres canales: 1) Rendición de cuentas 2) negociación de recursos y 3) Instrucciones, en el caso de un Equipo de SCRUM estas funciones se distribuyen en diferentes elementos del Marco SRUM:

S3-MS

El MS desempeña varias funciones dentro del S3 las cuales son:

- El SM es el elemento integrador de un equipo SCRUM, ya que es responsable de propiciarla la integración sinérgica y el funcionamiento armónico del equipo SCRUM; SM logra esto asegurándose de que el marco SCRUM, sus valores y los principios ágiles sean entendidos y aplicados por todos de la mejor manera posible a través del coaching.
- El SM es la única figura que cuenta con la visión de lo que está sucediendo con el equipo.
- Se encarga de velar como su fuese un escudo del equipo al proteger al equipo de impedimentos que puedan interferir con el trabajo del equipo durante la ejecución de cada iteración (Presente) como, por ejemplo: la introducción de nuevos requisitos, el secuestro de un miembro DT, la interferencia directa de las partes interesadas.
- El MS es el canal para llegar hacia el equipo de desarrollo esto a través del "canal de instrucciones del S3" por lo que el MS actúa de enlace entre el DT y el PO
- El MS se encarga de gestionar recursos que necesita el equipo para su correcto funcionamiento, ejemplos de ello son: Cuando el equipo requiere una nueva licencia de software el SM es responsable de gestionar esta necesidad con la figura correspondiente dentro de la organización. Otro

ejemplo es la introducción de cambios, aunque el PO tiene la última palabra en estos casos, el SM puede actuar como intermediario conciliador entre el equipo de desarrollo y el DP.

- **S3 Pila del sprint**

Este artefacto es el principal mecanismo de rendición de cuentas de los miembros del equipo de desarrollo, en dicho artefacto se encuentran las tareas por realizar, el equipo se encarga de asignar las tareas, por lo que cada tarea tendrá un responsable y este se encargará de actualizar el estado de la tarea (Por ejemplo: terminada, pendiente o en curso) e ir actualizando el esfuerzo restante de la misma cada día. Por lo que el este artefacto forma parte del S3 y utiliza el canal de rendición de cuentas.

3.3.2.4 Sistema 3* Auditoria

- **S3* Asistencia de otros interesados a las reuniones de SCRUM**

En las reuniones de prescritas por SCRUM, existe la posibilidad de que otras figuras no pertenecientes a la estructura de roles definida por SCRUM acudan a presenciar las reuniones, sin embargo, esto no significa que puedan intervenir directamente. Esto hace referencia al principio de transparencia de SCRUM y es parte del canal del S3* ya que puede obtener información que no podría obtener del equipo de desarrollo.

3.3.2.5 Sistema 2 Coordinación

El sistema dos se compone de los mecanismos de coordinación, comunicación y solución de problemas. Dichas funciones están distribuidas en los siguientes elementos del Marco SCRUM: MS, Tablero SCRUM, La técnica de caja de tiempo y la reunión diaria de SCRUM.

- **S2-MS**

El MS tiene un rol dentro de S2 ya que es responsable de facilitar todos los eventos que SCRUM prescribe (planificación de Sprint, las reuniones diarias de SCRUM, la revisión del sprint y la reunión de retrospectiva) esto ayuda al equipo a coordinarse y a reducir las oscilaciones (como salirse fuera de los objetivos de una reunión).

- **Tablero S2-Scrum**

Un tablero SCRUM es una herramienta que ayuda al equipo de desarrollo a comunicar en que está trabajando actualmente cada quien y cuál es su progreso. Esto ayuda al equipo SCRUM a conocer que es lo que está pasando para coordinar el trabajo, además de servir de fuente de información para identificar cuellos impedimentos como por ejemplo los cuellos de botella.

- **S2- Técnica de cajas de tiempo (time-boxing)**

El propósito de las cajas de tiempo (*timeboxing*) es definir y limitar la cantidad de tiempo dedicado a una actividad, SCRUM usa esta técnica para todos sus eventos y como una herramienta para definir específicamente tareas abiertas o ambiguas. Esta técnica ayuda a evitar perder el objetivo de las reuniones y reducir el desperdicio de tiempo, lo que lo convierte a esta técnica en un mecanismo anti oscilatorio del S2.

- **S2- Reunión de SCRUM diario**

Esta reunión diaria es un mecanismo de comunicación y coordinación que forma parte de S2, ya que este mecanismo ayuda al equipo a sincronizar el trabajo y establecer el plan para las próximas 24 horas; además, en esta reunión es cuando se actualiza el tablero de SCRUM además de ser un tiempo en donde es posible agregar los impedimentos identificados a la lista de impedimentos (para evitar oscilaciones).

3.3.2.6 Sistema 1 Operación

- **S1-Miembros del equipo de desarrollo**

El S1 es la parte operativa del Sistema en foco, este está formado por un conjunto de unidades operativas que representan el propósito global del sistema es decir “lo que hace”. Un Equipo SCRUM construye software y sus unidades operativas (quienes construyen finalmente el producto de software) son los Miembros del equipo de desarrollo. Todos los anteriores sistemas le dan soporte a la operación con el fin de garantizar la entrega de dichos incrementos y el producto final en el menor tiempo posible y con la máxima calidad posible para alcanzar la satisfacción del cliente.

3.4 Coherencia de la recursión del sistema

Para analizar la coherencia entre los diferentes niveles de recursión, nos enfocaremos en la esfera de recursión que identificamos en la Figura 18. En el criterio: “Project” podemos apreciar que este comienza con el proyecto ágil (L1) que es ejecutado por un ST (L2) integrado por un SM, un PO y el DT (L3) que está conformado por los profesionales que finalmente desarrollan el producto: los TM (L4). La coherencia y consistencia entre todos los niveles por lo tanto podemos describirla a través del Framework de SCRUM con el uso de los roles, artefactos y reglas asociadas.

Comenzando por el I S5, todos los S5 están conectados y gobernados por los Valores y Principios de SCRUM así como por la definición de SCRUM a través de todos los niveles de recursión. El S4 (el órgano de adaptación) de los sistemas, de igual forma está representado en todos los niveles a través del PO y artefactos como el PB, además el Homeostato S4-S3 está cubierto por la colaboración entre el PO y El SM. El Sistema 2 (la coordinación y la comunicación) es cubierto por

funciones de los roles PO y SM y también con los eventos con objetivos y reglas específicas además de los sistemas de información que pudieran usarse.

Finalmente, el S1 para el que podemos ubicarnos en el Sistema en foco en donde se puede apreciar la colaboración conjunta entre el PO, SM con el DT (Sistema en foco) representada por las líneas punteadas en la Figura 18, es en este lugar donde se aprecia el propósito del DT (no de la organización): “hace software”.

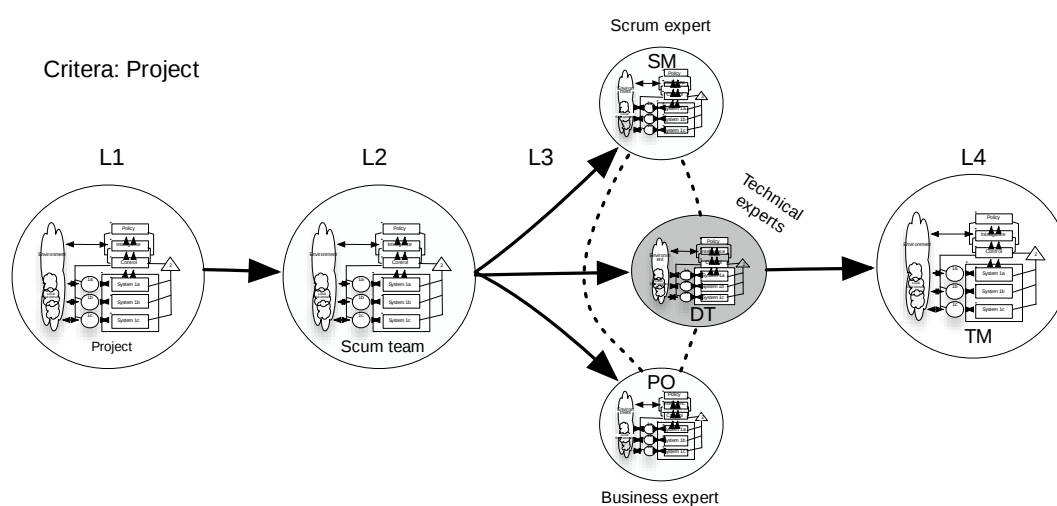


Figura 18 Esfera de recursión del criterio: Proyecto

Para que el DT pueda cumplir con su propósito (Éxito del proyecto) se requiere de la presencia y correcto funcionamiento de todos los sistemas a través de todos los niveles de recursión. Entonces la consistencia puede apreciarse por la presencia del marco SCRUM en todos los niveles de recursión que identificamos, que conecta y especifica funciones propias de las cinco funciones de viabilidad que hacen trabajar al equipo como un todo sistema autónomo

Sin embargo, es importante mencionar a la Organización entera nivel de recursión: 0 Presente en ambos criterios de recursión (en el criterio: "Project" se omite). Ya que es esta en donde un ST estará embebido como un S1, por ejemplo, en un “Departamento” que requiere de desarrollar un Proyecto de

Software, o si la organización es una "casa de desarrollo de software" el lugar donde el equipo esta embebido variara. Nosotros definimos una estructura organizacional genérica que empieza con la organización entera y termina con la persona (en el criterio: *Structure*).

La naturaleza de las organizaciones varia de muchas formas, por ejemplo, en la cultura organizacional, en la forma de "hacer las cosas". Todo esto tendrá influencia un ST independientemente del nivel de recursión. Estas influencias pueden beneficiar los propuesto por SCRUM pero también pueden afectar la esencia de SCRUM; esto último, especialmente en organizaciones de administración tradicional.

Es por esta razón en la que se eligió el criterio de "proyecto" ya que nos permite concentrarnos en lo que SCRUM propone y además nos habilita a poder omitir la dependencia a la naturaleza de la organización. Si tenemos en cuenta que: el marco SCRUM define tres elementos indispensables que deben de estar presentes en cualquier implementación de SCRUM, estos son: Los roles, eventos y artefactos como se muestra en la Figura 10, si cualquiera de estos llegase a faltar el resultado sería cualquier otra cosa y no SCRUM.

Es por ello que, más arriba del nivel del ST puede haber incoherencias provenientes de la organización en la que el ST este embebido, por ejemplo, si la organización no está alineada con las formas de hacer las cosas en SCRUM, sin embargo, esto último sale fuera del propósito de esta tesis.

Capítulo 4 Modelo basado en Agentes de un Equipo SCRUM

A partir del Modelo de Sistema Viable del equipo de desarrollo SCRUM el siguiente paso es llevar dicho modelo a un modelo computacional, el enfoque seleccionado es el Modelado Basado en agentes y la simulación. Ambos enfoques fueron utilizados en conjunto de forma complementaria como se muestra en la Figura 19.

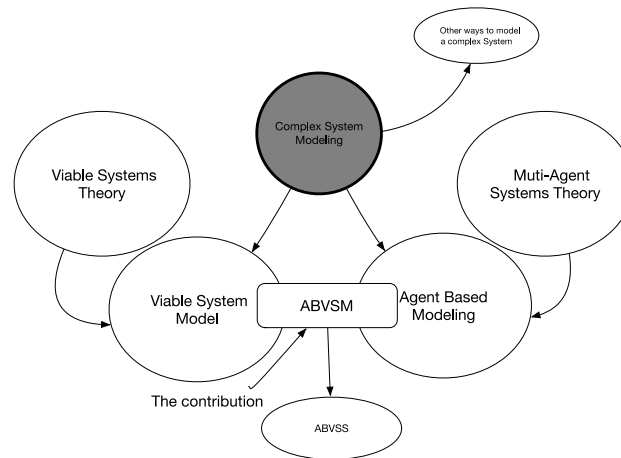


Figura 19 VSM y ABMS

A demás de lo que hemos analizado con el Modelo de sistemas viables, hemos considerado los factores de desempeño descritos en [38] y [39] y también un estudio sobre factores de impacto negativo [7] para el diseño del modelo basado en agentes.

Uno de los factores que se considera en este artículo son los rasgos de personalidad de los integrantes de un equipo, ya que diversos estudios sostienen que dichos rasgos influyen en cómo se comportan las personas en el contexto de trabajo. Además, Estudios como [40], [41] sostienen que dichos rasgos están relacionados con la aptitud para desempeñar determinadas tareas o roles.

El modelo conceptual es extraído del Modelo de Sistema viable. Este proceso se ilustra en la Figura 20.

Identificación de los Agentes

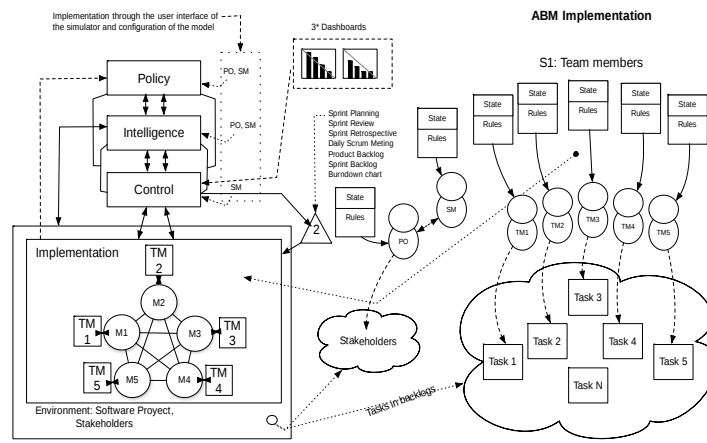


Figura 20 Proceso del VSM a ABM

Agentes:

- Agente *Team Member*.

-Atributos: características propias (experiencia, rasgos de personalidad, estado de ánimo).

-Reglas de comportamiento: actividades del rol (Desarrollo del producto, organización del trabajo, reporte de progreso), interacción con otros agentes.

- Agente *SCRUM Master*

-Atributos: características propias (experiencia, rasgos de personalidad, estado de ánimo).

-Reglas de comportamiento: actividades del rol (Enseñar SCRUM, administrar el proceso, proteger al equipo, reforzar las reglas, remover obstáculos).

- Agente *Product Owner*.

-Atributos: características propias (experiencia, rasgos de personalidad, estado de ánimo).

-Reglas de comportamiento: actividades del rol (Priorización de las características del producto, observar y aconsejar).

Medio ambiente:

El medio ambiente está compuesto por el proyecto y los artefactos de SCRUM (*product backlog*, *Sprint backlog* y el incremento). El proyecto se conforma por las historias de usuario, las cuales son elementos en los que se divide todo el trabajo que el equipo tiene que hacer en un proyecto SCRUM; dichas historias se fraccionan en tareas que el equipo se auto-asigna para realizarlas en cada sprint. En el modelo estas tareas se especifican de la siguiente manera:

- *Tareas*

-Atributos: características de la tarea (prioridad, complejidad, sprint, tipo, estimación, progreso, estado actual, asignada-a)

Interacciones:

Las interacciones consideradas en el modelo se muestran en la Figura 21 y son las siguientes:

- 1.- Interacción *team member* – *team member*
- 2.- Interacción *team member* – ambiente
- 3-4.- Interacción *team member* – *SCRUM Master*
- 5.- Interacción *SCRUM Master* – ambiente
- 6.- Interacción *product owner* – ambiente
- 7-8.- Interacción *SCRUM Master* – *product owner*

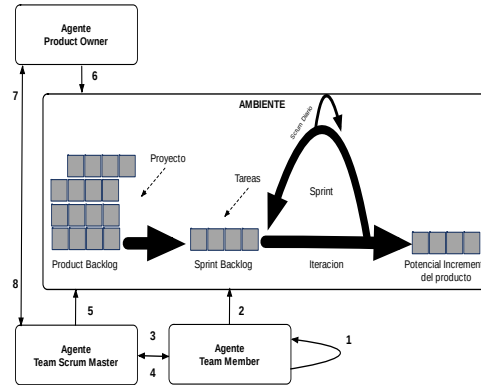


Figura 21 Interacciones en la dinámica de SCRUM

Representación

A continuación, se presenta la formalización del modelo conceptual en un ABMS. En la Tabla 3 se muestra la representación de las entidades del modelo (Roles SCRUM), los roles SCRUM Master y *Product Owner* y la Tabla 4 y el medio ambiente (proyecto), así como los rangos de los atributos de dichos componentes.

Tabla 3 Representación de los atributos de los agentes

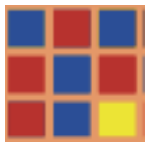
Agentes	Atributos	Representación
Team Member 	Personalidad: Extraversión, Apertura a la experiencia, Responsabilidad, Afabilidad Neocriticismo	Baja (20 - 45) Media (46 - 65) Alta (65 - 100)
	Experiencia, Afinidad (Ana, Dev, Prog, Test, Pres)	Alta (.67 – 1) Media (.34 - .66) Baja (0 - .33)
	Estado de animo	Positivos – negativos
	Probabilidad de error en una tarea	0 - 1
	Avance promedio	0-100%
SCRUM Master y Product Owner	Personalidad: Extraversión, Apertura a la experiencia, Responsabilidad, Afabilidad Neocriticismo	Baja (20 - 45) Media (46 - 65) Alta (65 - 100)
	Experiencia, habilidades	Alta (.67 – 1)

		Media (.34 - .66) Baja (0 - .33)
	Estado de animo	Positivos – negativos

En la **¡Error! No se encuentra el origen de la referencia.** se muestra los rangos para los atributos del agente *Team Member*. La escala de valores para el perfil de personalidad son tomados de la teoría de personalidad de los cinco grandes factores [42], el atributo estado de ánimo se basa en el trabajo de [43]. Se utilizó una escala propia definida empíricamente para los aspectos de experiencia.

El medio ambiente se representa en la Tabla 4 Ya que los proyectos en SCRUM utilizan un ciclo iterativo e incremental, el medio ambiente se definió en *sprints* y las tareas, los rangos son descritos a continuación.

Tabla 4 representación de los atributos del medioambiente

	Atributos	Representación
Sprint	Numero de sprints	1-10
	Tamaño del sprint	1 – 30
	Numero de tareas	1 - 100
Tareas 	Progreso de avance	0-100%
	Progreso de prueba	0-100%
	Numero de sprint	0-12
	Complejidad	Baja, Media, Alta
	Prioridad	Baja, Media, Alta
	Tipo	Ana, Prog, Doct, cord.
	Estimación (puntos)	0 - 100
Proyecto	Probabilidad de que surja una nueva tarea	0-1

A continuación, se muestra un diagrama de clases del modelo basado en agentes:

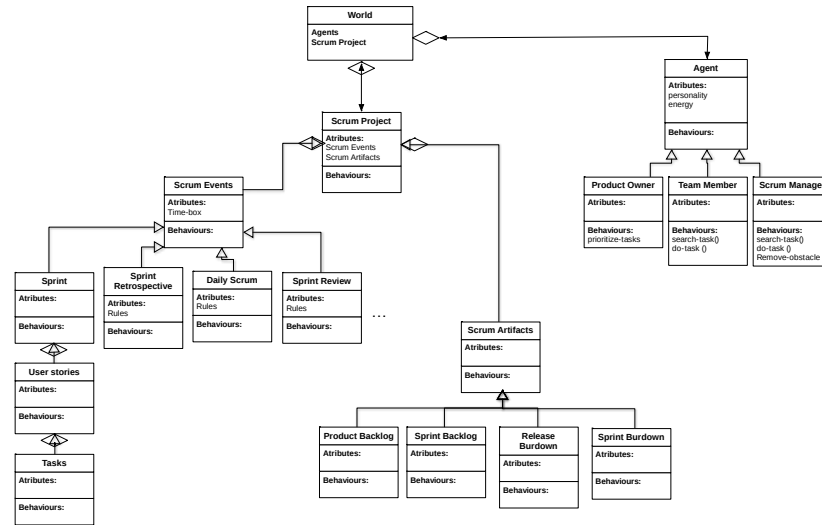


Figura 22 Diagrama de clases de SCRUM

Capítulo 5 Caso de Estudio

Desarrollo del simulador

Se desarrolló una primera implementación del modelo en un prototipo de simulador, programado en la herramienta Netlogo versión 5.3.1 que utiliza un lenguaje de programación inspirado en Logo [44]. En dicho prototipo, las funciones del SCRUM Master y el *Product Owner* fueron implementadas por medio de la interface de usuario, además, en esta primera versión se enfocó en el aspecto del desempeño del equipo, por lo que el aspecto de la estimación del proyecto no fue incluido.

Interface del simulador

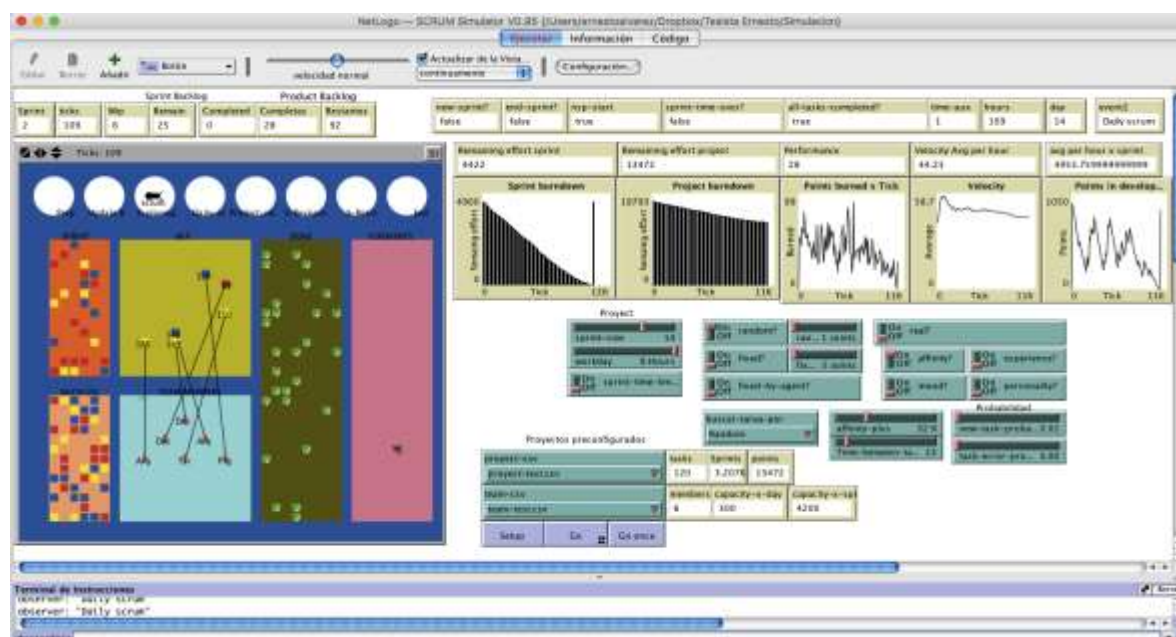


Figura 23 Interface del simulador en netlogo

Para el diseño de la interface se optó por dividirla en zonas como se muestra en la **¡Error! No se encuentra el origen de la referencia.**, las zonas son: zona 1 control, zona 2 parámetros, zona 3 social, zona 4 medio ambiente y zona 5 monitoreo.

Zonas del Simulador

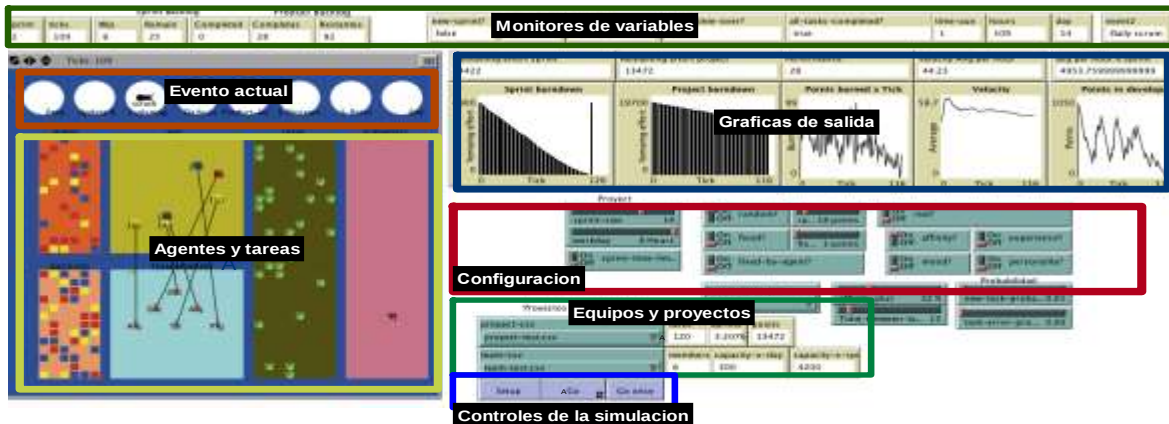


Figura 24 Zonas del simulador

Zona: Configuración

En esta zona se encuentran los controles para configurar la simulación y para definir los valores iniciales de las variables del modelo con respecto a la configuración del proyecto, los aspectos a considerar de los agentes, y la forma de asignación de tareas.

Zona: Controles de la simulación

Contiene tres botones para controlar la ejecución de la simulación: el botón *Setup* el cual tiene la función de inicializar los parámetros de las variables del simulador, el botón *Go* tiene la función de ejecutar la simulación hasta que el proyecto es completado por los agentes y finalmente el botón *Go once* el cual ejecuta la simulación un tiempo (un tick).

Zona: Equipos y proyectos

En esta zona se puede seleccionar los equipos y los proyectos, que están previamente configurados, estos equipos son cargados de archivos CVS configurados con las características de los equipos y los proyectos.

Zona: Monitoreo de variables

En esta zona se muestran las variables de la simulación referentes a los aspectos de SCRUM, el proyecto y los agentes, estos van variando a lo largo de la simulación.

Zona: Graficas de salida

En esta zona se muestran las gráficas de salida, en los que se puede apreciar el comportamiento de la simulación, Es en esta zona donde se puede apreciar los artefactos como la pila del producto y la pila del sprint, así como variables de desempeño del equipo.

Zona: Agentes y tareas

En esta zona se muestra la interacción de los agentes con el proyecto (las tareas), es la parte que refleja la parte colaborativa de la simulación. Esta zona contiene seis carriles que se asemejan a un tablero SCRUM. Dichos carriles son: el product *backlog*, el sprint backlog, la sección de trabajo en progreso (WIP: *work in progress*), el carril *Team Members*, el carril de hecho (done) y finalmente el carril de incrementos. Los colores de los agentes representan su estado de ocupado o desocupado y en las tareas el color refleja la prioridad, mientras que los enlaces representan la relación de los agentes con la tarea que se encuentran realizando, los enlaces pueden ser normales y enlaces fuertes dependiendo de la afinidad a la tarea, esto puede apreciarse en el grosor del enlace.

Zona: Evento actual

En esta zona se muestra el evento actual de SCRUM que está teniendo lugar en la ejecución de la simulación, esto se puede apreciar de acuerdo a la posición del agente SCRUM en los círculos en blanco que simbolizan los eventos de SCRUM.

Finalmente, la zona 5 es donde se monitorea el comportamiento de la simulación a través de las gráficas de salida. Para este prototipo de simulador se programaron dos: (a) la velocidad promedio de quemado (burndown) y (b) el desempeño global de los agentes, la escala de tiempo que se maneja en el prototipo son ticks.

Capítulo 6 Resultados

La investigación inició tomando como partida la especificación del Caso de Estudio como referencia a los equipos de desarrollo de software bajo la metodología ágil conocida como SCRUM.

Primeramente se describe la naturaleza y la identidad de los equipos SCRUM, dicha descripción tiene el objetivo de dejar claro cuál es el propósito del sistema es decir ¿cuál es el propósito de los Equipos SCRUM?, además, se describen los aspectos que tienen influencia para el propósito del mismo.

Siendo los equipos SCRUM una organización que esta embebida dentro de otra organización, se utilizó el Modelo de Sistemas Viables como herramienta para modelar y analizar la viabilidad de los Equipos SCRUM, objetivo principal de nuestra investigación.

Esta etapa tuvo el objetivo de entender el funcionamiento de los equipos SCRUM al abstraer los aspectos organizacionales que hacen posible que los equipos de SCRUM sean viables.

Así, partiendo del modelo de Sistemas Viables del equipo, se creó un modelo basado en agentes. En este modelo se identificaron y especificaron los agentes, su medio ambiente, así como las reglas de interacción asociadas. Dicho modelo fue la base para el desarrollo de un prototipo en la siguiente etapa y el establecimiento de un experimento para obtener resultados cuantitativos.

A continuación se desarrolló un prototipo de simulador que utiliza el modelo basado en agentes creado a partir del modelo de Sistema Viable del Equipo SCRUM. Se utilizó la plataforma Netlogo en su versión 5.1.3.

En esta plataforma se realizó un experimento para comparar el desempeño de un equipo SCRUM en el aspecto de la asignación de tareas, definiendo tres escenarios: (1) Asignación aleatoria, (2) Asignación por afinidad, (3) asignación por prioridad. Al analizar los datos arrojados por el prototipo del simulador, se contrastan los resultados de los tres escenarios para obtener las conclusiones.

Validando el simulador con un panel de expertos en SCRUM recogiendo las impresiones del prototipo del simulador ante miembros de equipos de desarrollo de software.

Se reunió a un grupo de cinco expertos certificados en SCRUM (dos académicos, dos programadores y un instructor). Se lograron dos objetivos, el primero, fue realizar una validación inicial del modelo y segundo, fue conocer la percepción acerca del primer prototipo de simulador.

La reunión se dividió en dos partes. En la primera parte, se les pidió a los expertos que en equipo discutieran y determinaran a su juicio cuáles son las características que debería de contemplar un modelo de SCRUM. Además de los requerimientos deseables en un simulador de SCRUM. En la segunda parte, se les pidió a los expertos que validaran el modelo y evaluaran nuestro prototipo de simulador, para ello se utilizó la lista de características y de requerimientos que previamente propusieron. Tanto la validación como la evaluación se realizó mediante un cuestionario con una escala del Likert de cinco niveles normalizados. En la Figura 25 se muestran los resultados.

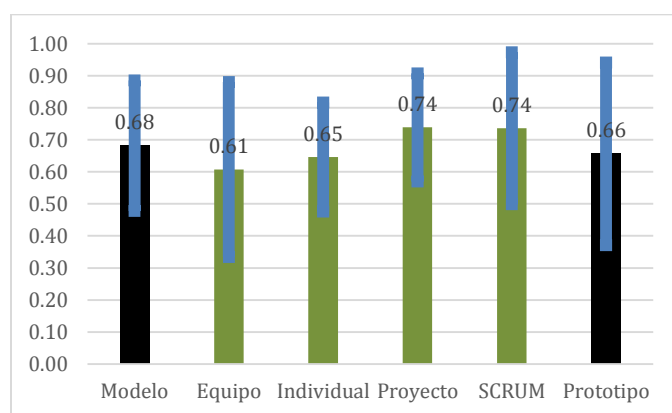


Figura 25 Resultados de la validación inicial con el grupo de expertos

Las categorías con un índice de validación más bajo fueron: Equipo e Individual con 0.61 y 0.65 respectivamente. Este resultado posiblemente se debió a: primero a las características que no estaban presentes en el modelo y segundo a aquellas que no estaban implementadas de la forma que los expertos propusieron. Por ejemplo, en el primer caso el modelo no contemplo la calificación

obtenida de los agentes en las tareas. En el segundo caso, la propiedad de la experiencia en los agentes en el modelo se definió como estática mientras que los expertos la definieron dinámica.

Por otro lado, Las categorías mejor evaluadas fueron: Proyecto y SCRUM ambas con 0.74. Esto se puede interpretar de que el modelo logra caracterizar SCRUM y los proyectos en un grado aceptable, aunque existe un espacio de mejora.

El modelo obtuvo un índice de validación de 0.68, lo cual quiere decir, una apreciación ligeramente superior al punto medio de aceptación. Consideramos que esta es una aproximación aceptable para el modelo inicial y su primera implementación en un prototipo de simulación. A su vez existe un margen para mejorar tanto el modelo como el simulador que puede alcanzarse agregando las características que el modelo no considera que los expertos mencionaron de acuerdo a su experiencia práctica en SCRUM.

SIMULACION DE TRES ESCENARIOS

A continuación, se describen brevemente tres ejemplos de escenarios de simulación con el prototipo: escenario (1) un caso ideal; escenario (2) considerando un factor de error en las tareas, y escenario (3) el caso donde se considera la influencia de los rasgos de personalidad y las habilidades y experiencia de los agentes. Los tres escenarios comparten los siguientes parámetros iniciales: número de *sprints*: 6; tamaño del equipo: 6; número de tareas por sprint: 20.

En el primer escenario no se considera la influencia de los factores experiencia y personalidad sobre el desempeño en las tareas, además, éstas siempre son terminadas con éxito y no existe la posibilidad de un error, en la

Figura 26 se muestran los resultados de este escenario. En las gráficas de salida se puede observar un avance constante en cada sprint, éste comportamiento ideal también se ve reflejado en la gráfica del proyecto. Además, puede observarse que los indicadores de desempeño y velocidad son estables. Aunque en la práctica estos escenarios podrían ser poco frecuentes, ya que la realidad muchas veces no se llega a lo ideal, intervienen múltiples factores, los cuales son considerados en los siguientes escenarios.

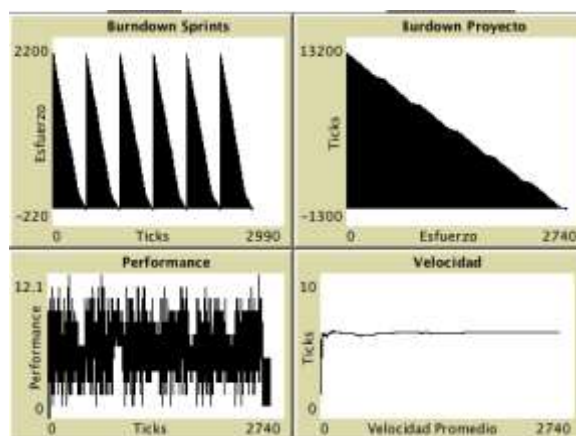


Figura 26 Caso ideal

En el segundo escenario se considera el factor error en las tareas, es decir, se contempla la posibilidad de que un integrante de un equipo falle en una tarea o ésta tenga algún error. En SCRUM al completar una tarea ésta puede pasar por una fase de prueba (*test*), en la interface del simulador esta etapa se representa por el carril *test* del tablero de SCRUM del prototipo, cuando una tarea entra en este carril significa que está siendo probada; si pasa la prueba la tarea es completada y se mueve a la zona *done*, en caso contrario la tarea cambiará al estado de *error*. Así en el prototipo cuando esto sucede la tarea regresará arbitrariamente al carril del *work in progress (wip)* con el progreso obtenido antes de que ocurriera el error.

La posibilidad de error puede ser controlada en la interface manipulando el control tipo deslizador *error rate* mientras mayor sea esta frecuencia de error, mayor será la posibilidad de que los agentes fallen en las tareas fallen o

contengan errores, en la Figura 27 se muestran las gráficas de quemado por *sprints* y global (proyecto) variando el parámetro *error rate* con los siguientes casos: caso 1 error de 0.3, caso 2 error de 0.6, caso 3 error de 0.9.

Con un error de 0.3 se aprecia un comportamiento parecido al escenario ideal, con un error de 0.6 y 0.9 los saltos en el comportamiento de las gráficas de quemado son más visibles. Se observa que la cantidad de *ticks* para que los agentes completen con éxito las tareas se dispara si el valor del error es mayor. Con un error de 0.3 se requirió de 4143 *ticks*; con un error de 0.6 de 5228 *ticks* y con un error de 0.9 de 17615 *ticks*.

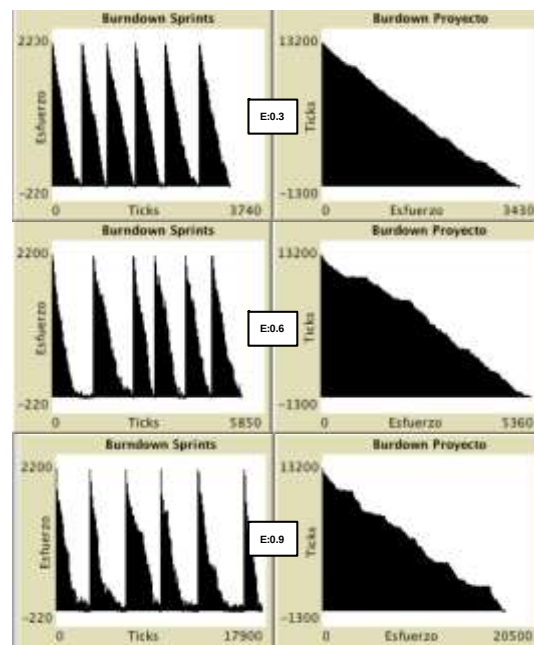


Figura 27 Gráfica de quemado considerando el error en las pruebas

Por último, En el tercer escenario se considera un error de 0.3 además de los rasgos de personalidad del *Big Five*. Se programaron reglas en los agentes para los factores Extraversión y Responsabilidad, basados en que estos factores están relacionados con la aptitud para la ejecución de las tareas de tipo desarrollo [41], es decir las tareas que involucran codificación.

En Figura 27 se observa el resultado en las gráficas de salida, en ellas se puede destacar los brincos que ocurrieron en la gráfica de velocidad del equipo,

estos se debieron a que en ese momento algún o algunos agentes escogieron tareas para los que son más aptos, un ejemplo de esto se representa con agente conectado a una tarea mediante un línea gruesa que simboliza una mayor aptitud hacia este tipo de tarea.

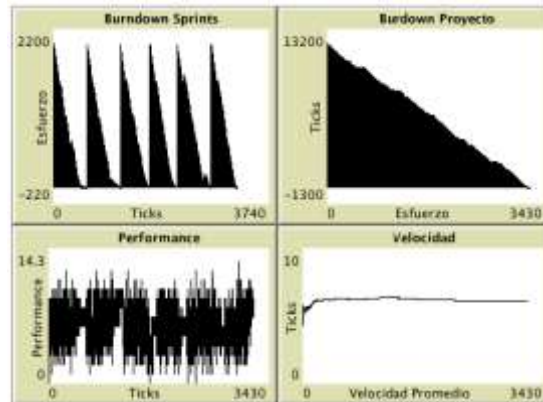


Figura 28 Gráficas considerando rasgos de personalidad para la aptitud de programación

En el prototipo se miden dos indicadores de desempeño del equipo, y se utilizan como variables de salida. dichos indicadores siguen un enfoque *burndown* (trabajo consumido) utilizado en SCRUM. Estas variables de salida se describen a continuación:

$$DE = \sum_0^n ETA \quad \dots\dots\dots (1)$$

$$VE = \frac{\sum_0^n ETA[n] - \sum_0^n ETA[n-1]}{n} \quad \dots\dots\dots (2)$$

Donde:

DE: desempeño del equipo por *tick*.

ETA: Esfuerzo total por *tick* de los agentes.

VE: velocidad del equipo.

n: número de *ticks*.

SIMULACION BASADA EN LA AFINIDAD DE TAREAS

En el experimento dos se realizó enfocándose en la Afinidad a las tareas de los miembros del equipo, las siguientes tablas describen el experimento.

Tabla 5 Configuración de la simulación

Configuracion de la simulacion	
Tamaño del sprint	14 días
Jornada laboral	8 horas
Incremento por afinidad	20 %
Avance de los agentes fijo	20 puntos
Rango de avance aleatorio de los agentes	0-20 puntos
Probabilidad de nuevas tareas	10%
Rango de tareas nuevas que pueden emerger	0-5
Respetar <i>time-boxing</i> del sprint	encendido
Complejidad tareas Baja	100 puntos
Complejidad tareas Media	150 puntos
Complejidad tareas Alta	200 puntos

Tabla 6 Configuración del equipo de desarrollo

Equipo de desarrollo		
Team member	Afinidad	Experiencia
1	Analisis	Alta
2	Programacion	Alta
3	Programacion	Alta
4	Validacion	Alta
5	Documentacion	Alta
6	Coordinar - Supervisar	Alta

Tabla 7 Configuración del proyecto

		Proyecto A
Complejidad* de la tarea	Alta	10
	Media	10
	Baja	10
Tipo de tarea	Análisis	6
	Programación	6
	Validar	6
	Documentar	6
	Coordinar-Supervisar	6

*Tamaño de la tarea en puntos

Tabla 8 Resultados Experimento 2

Resultados de las simulaciones								
Proyecto A								
Caso	Tipo de asignación	Progreso	Corridas	Número de sprints	Velocidad promedio	Puntos en desarrollo promedio	Desempeño puntos quemados por tick	Horas
1	Selección aleatoria	Fijo	100	1.92	51.98	353.79	43.98	210
		Aleatorio	100	3.18	25.69	450.76	24.49	357
2	Selección por afinidad	Fijo+afinidad	100	2.2	33.09	467.37	29.81	290.11
		Aleatorio+afinidad	100	3.89	21.31	373.98	20.09	436

El caso ideal el que fue hecho en menor tiempo fue la selección aleatoria con progreso fijo, este escenario supone que todos los agentes progresan lo mismo, sin embargo dicho caso no está apegado a la realidad, si lo tomamos como referencia al considerar la afinidad fijo más afinidad es el caso que más se acerca a esta referencia, por lo que se puede apreciar que el considerar la afinidad para la asignación de resultados da un desempeño cercano al ideal.

Capítulo 7 Conclusiones

Los equipos SCRUM son un tipo de equipo de trabajo que ha tomado auge actualmente basado en metodologías ágiles para el desarrollo de software. Estos equipos trabajan bajo un marco de trabajo que prescribe roles, eventos, artefactos y reglas asociadas. En nuestra investigación el primer paso para llegar a un modelo computacional que simule la interacción y actividades del equipo SCRUM fue entender cómo es que funcionan realmente los equipos SCRUM. Para lograr esto, se optó por el Modelo de Sistemas Viables el cual nos permitió modelar formalmente al Equipo SCRUM.

Con SCRUM podemos ver en práctica mucho de los principios presentes en el enfoque de los Sistemas Viables. Entre los que destacan la Auto organización, la comunicación, la colaboración. Además comparten un estilo de gestión con un alto grado de autonomía confiando en el conocimiento tácito de las personas en lugar de el comando y control. Los Roles de SCRUM no poseen jerarquía unos con otros siendo esto compatible con la filosofía del Modelo de Sistemas Viables al no tener exceso de niveles jerárquicos.

El Modelo de Sistemas Viables del Equipo SCRUM nos permite entender el rol de cada uno de los componentes que el marco de trabajo de SCRUM prescribe. Esto se logró mediante un mapeo de dichos componentes con las cinco funciones de viabilidad (sistema 1 a 5). El sistema 1 son los miembros de equipo de desarrollo que cubren la operación del equipo SCRUM; el sistema 2 son los artefactos y reuniones siendo estos los que ayudan a la coordinación del equipo; el sistema 3 es el SCRUM Master, la pila del sprint que llevan el control del equipo; el sistema 4 es el Product Owner junto con el SCRUM Master, las revisiones del sprint, retrospectiva y pila de producto que son la parte central de inteligencia del equipo y trabajo de desarrollo; y el sistema 5 son las metas del sprint, definición de hechos, visión y teoría de SCRUM, el manifiesto ágil, valores y principios de SCRUM ya que todos ellos son las políticas con las cuales se rige en equipo de trabajo de nuestro caso de estudio.

Para llevar este modelo a un modelo computacional se recurrió al Modelado Basado en Agentes, esta metodología por su flexibilidad nos permitió implementar el Modelo Basado en Agentes de un Equipo SCRUM en un prototipo de simulador, donde nos enfocamos únicamente en un caso de estudio donde se abordó el aspecto de la asignación de tareas.

El escenario que se acerca a lo más ideal en la realidad para trabajar eficientemente, es aquel en que las tareas se asignan por afinidad, en este caso el equipo de trabajo tiene un mejor desempeño. Habrá menos desempeño cuando la asignación se realiza de forma aleatoria. Es por esto que el simulador ayuda a dar una estimación mejor en la realidad si se asignan las tareas de acuerdo a las capacidades y afinidades del miembro del equipo.

Todo esto nos lleva a concluir que los equipos de SCRUM pueden ser modelados computacionalmente utilizando el enfoque de Sistemas Viables y el Modelado basado en Agentes y la Simulación.

El proponer un modelo de un equipo de desarrollo de software desde el enfoque MVS creemos que nos permitirá crear equipos viables, tomando la definición de viable como “capaz de mantener una existencia separada”. Esto lo interpretamos en que el equipo podrá poseer la autonomía suficiente para trabajar de manera efectiva, poseerá las capacidades de adaptación y reacción a cambios inesperados que atenten contra la supervivencia del equipo, además permite definir una estructura que puede ser utilizada para futuros estudios científicos.

Nuestro prototipo permite probar equipos de trabajo antes de llevar a la producción real. Así mismo permite estimar el desempeño posible de un equipo y el tiempo en que tardaría en completar un proyecto. Y ayuda a integrar de una mejor forma los equipos de trabajo.

Como trabajo a futuro se recomienda realizar pruebas en organizaciones reales e implementar el modelo basado en agentes en un simulador que integre asignación de tareas, interdependencia de tareas, roles, personalidad y estados de ánimo de los miembros del equipo. Como área de oportunidad a investigar es modelar otros comportamientos emergentes del equipo como lo son conflictos y cambios inesperados.

BIBLIOGRAFÍA

- [1] A. Fuggeffa, A. Fuggetta, and P. Milano, "Software Process : A Roadmap Software Process : A Roadmap," vol. 97, 1988.
- [2] K. M. Whitney and C. B. Daniels, "The root cause of failure in complex IT projects: Complexity itself," *Procedia Comput. Sci.*, vol. 20, pp. 325–330, 2013.
- [3] M. Zanoni, F. Perin, F. A. Fontana, and G. Viscusi, "Complex software project development: agile methods adoption," *J. Softw. Evol. Process*, vol. 26, no. 12, pp. 1172–1192, 2014.
- [4] R. Victor, "IIDHistory," no. June. pp. 47–56, 2003.
- [5] VersionOne.com, "11th Annual State of Agile Report," 2017.
- [6] Scrumalliance.org, "the State of Scrum Report," 2017.
- [7] V.-P. Eloranta, K. Koskimies, and T. Mikkonen, "Exploring ScrumBut—An empirical study of Scrum anti-patterns," *Inf. Softw. Technol.*, vol. 74, pp. 194–203, 2016.
- [8] M. Yilmaz and R. V. O. Connor, "A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study," *Teh. Vjesn. - Tech. Gaz.*, vol. 23, no. 1, pp. 237–245, 2016.
- [9] S. Sawyer, "Software development teams," *Commun. ACM*, vol. 47, no. 12, pp. 95–99, 2004.
- [10] S. Wolfram, *A new kind of science*, vol. 5. Wolfram media Champaign, 2002.
- [11] J. I. G. V Medina, "Agent-based Modelling: A New Way of Exploring Social Phenomena," *Rev. Esp. Investig. Sociol.*, pp. 91–109, 2011.
- [12] G. Rodriguez, A. Soria, and M. Campo, "Supporting virtual meetings in distributed scrum teams," *IEEE Lat. Am. Trans.*, vol. 10, no. 6, pp. 2316–2323, 2012.
- [13] Y. Lin, P. Descamps, N. Gaud, V. Hilaire, and A. Koukam, "Multi-Agent System for intelligent Scrum project management," *Integr. Comput. Aided. Eng.*, vol. 22, no. 3, pp. 281–296, 2015.
- [14] J. Domann, S. Hartmann, M. Burkhardt, A. Barge, and S. Albayrak, "An agile method for multiagent software engineering," *Procedia Comput. Sci.*, vol. 32, no. 0, pp. 928–934, 2014.
- [15] S. Beer, *Brain of the Firm*. Chichester: John Wiley, 1972.
- [16] S. Beer, *The Heart of Enterprise*. Chichester: John Wiley, 1979.
- [17] S. Beer, *Diagnosing the System for Organizations*. Chichester: John Wiley, 1985.
- [18] R. H. Raul Espejo, Ed., "The viable System Model Interpretations and Applications of Stafford Beer's VSM," in *The viable System Model Interpretations and Applications of Stafford Beer's VSM*, John Wiley & Sons, 1989.
- [19] P. Hoverstadt, *The Fractal Organization: Creating sustainable organizations with the Viable System Model*. Wiley, 2011.
- [20] S. Beer, *Brain of the Firm*, 2nd ed. Chichester: John Wiley, 1995.
- [21] R. Espejo and A. Gill, "The viable system model as a framework for understanding organizations," *Phrontis Ltd. SYNCHO Ltd.*, 1997.

- [22] M. Schwaninger, "Design for viable organizations: The diagnostic power of the viable system model," *Kybernetes*, vol. 35, no. 7/8, pp. 955–966, 2006.
- [23] J. H. Holland, "Studying complex adaptive systems," *J. Syst. Sci. Complex.*, vol. 19, no. 1, pp. 1–8, 2006.
- [24] H. Sayama, *Introduction to the Modeling and Analysis of Complex Systems*. SUNY Geneseo, 2015.
- [25] A. Societies and S. Simulation, "Tina Balke and Nigel Gilbert (2014) How Do Agents Make Decisions ? A Survey Introduction : Purpose & Goals Dimensions of Comparison," vol. 17, 2014.
- [26] C. M. Macal and M. J. North, "Agent-Based Modeling and Simulation: AMBS Examples," *2008 Winter Simul. Conf.*, pp. 101–112, 2008.
- [27] N. Gilbert and K. G. Troitzsch, *Simulation for the Social Scientist*. 1999.
- [28] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation," *J. Simul.*, vol. 4, no. 3, pp. 151–162, Sep. 2010.
- [29] G. N. Gilbert, *Agent-based models*, no. 153. Sage Publications Inc., 2007.
- [30] U. Wilensky and W. Rand, *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. MIT Press, 2015.
- [31] J. Schwaber, Ken; Sutherland, "Scrum Guide," 2016. [Online]. Available: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf>. [Accessed: 05-Sep-2017].
- [32] I. Kayes, M. Sarker, and J. Chakareski, "Product backlog rating: a case study on measuring test quality in scrum," *Innov. Syst. Softw. Eng.*, vol. 12, no. 4, pp. 303–317, 2016.
- [33] Scrum-Alliance, "The 2015 State of Scrum Report," 2015.
- [34] T. O. A. Lehtinen, J. Itkonen, and C. Lassenius, "Recurring opinions or productive improvements???what agile teams actually discuss in retrospectives," *Empir. Softw. Eng.*, pp. 1–44, 2016.
- [35] J. M. P. Ríos, *Design and diagnosis for sustainable organizations: the viable system method*. Springer Science & Business Media, 2012.
- [36] R. Pichler, *Scrum: agiles Projektmanagement erfolgreich einsetzen*. dpunkt.verlag, 2013.
- [37] P. R. Jose, *Design and Diagnosis for Sustainable Organizations: The Viable System Method*. Springer Berlin Heidelberg, 2012.
- [38] E. Salas, J. W. Guthrie, K. A. Wilson-Donnelly, H. A. Priest, and C. S. Burke, "Modeling Team Performance: The Basic Ingredients and Research Needs," in *Organizational Simulation*, John Wiley & Sons, Inc., 2005, pp. 185–228.
- [39] R. C. Lee, "The Success Factors of Running Scrum: A Qualitative Perspective," *J. Softw. Eng. Appl.*, vol. 5, no. 6, pp. 367–374, 2012.
- [40] L. Martínez, A. Rodríguez-Díaz, G. Licea, and J. Castro, "Big Five Patterns for Software Engineering Roles Using an ANFIS Learning Approach with RAMSET," in *Advances in Soft Computing SE - 37*, vol. 6438, G. Sidorov, A. Hernández Aguirre, and C. Reyes García, Eds. Springer Berlin Heidelberg, 2010, pp. 428–439.
- [41] T. Gnambs, "What makes a computer wiz? Linking personality traits and programming aptitude," *J. Res. Pers.*, vol. 58, pp. 31–34, 2015.
- [42] M. R. Barrick and M. K. Mount, "The big five personality dimensions and job

performance: a meta-analysis," *Pers. Psychol.*, vol. 44, no. 1, pp. 1–26, 1991.

- [43] L. Miller, *Mood Mapping: Plot Your Way to Emotional Health and Happiness*. Rodale, 2009.
- [44] U. Wilensky, "Netlogo." Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, 1999.

Anexo 1 Encuesta prototipo

Cuestionario TAM

Proyecto: Simulador de SCRUM **Fecha:** 28 de marzo del 2017

Responsable: MTIC Ernesto Raúl Álvarez Molina, Estudiante de Doctorado en ciencias, Universidad Autónoma de Baja California.

Objetivo: Evaluar el Simulador de Scrum a través del modelo de aceptación tecnológica TAM. A través de preguntas donde se determina cinco variables: Facilidad de uso percibida, Utilidad percibida, Uso anticipado, actitud hacia el uso e intención del uso.

Nombre: _____ Lugar de trabajo: _____

Puesto: _____ Años de experiencia en su cargo: _____ Título Profesional más alto: _____

Instrucciones: marca con una X la opción que consideres apropiada. Por favor responde de la manera más honesta posible.

a. Facilidad de Uso Percibida

	Totalmente en desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	Totalmente de acuerdo
Podría ser fácil para mí volverme hábil en usar el simulador scrum					
Me sería posible utilizar el simulador scrum sin ayuda de un experto					
Encuentro al sistema fácil de usar					
Aprender cómo utilizar el simulador scrum es fácil para mí					
Se necesita poco tiempo para aprender a utilizar el simulador scrum					
Es fácil recordar cómo se usa el simulador scrum					
Usar el simulador scrum es fácil para cualquier persona					
Usar el simulador scrum requiere de poco esfuerzo mental					
En general, encuentro fácil el uso del simulador scrum					

b. Utilidad Percibida

	Totalmente en desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	Totalmente de acuerdo
Considero que las funciones del simulador scrum son útiles para mí					
El uso de un simulador scrum en mi trabajo me ayudaría a ser más efectivo en mi trabajo					
Usando el simulador scrum mejoraría el desempeño de mi equipo					
Utilizar un simulador de scrum me permite conocer el potencial de un equipo					
En general, encuentro el simulador scrum de gran utilidad					

c. 4. Uso anticipado del nuevo sistema

Para las siguientes preguntas, asuma que el simulador scrum se encuentra disponible para Usted y lo usaría para obtener éxito en su (futuro) trabajo.

1. Asuma que el nuevo sistema estaría disponible en su trabajo. Predigo que lo usaré sobre bases regulares en el futuro.

Imposible	Baja Posibilidad	Medianamente Posible	Altamente Posible
-----------	------------------	----------------------	-------------------

2. En mi trabajo la frecuencia con la que usaría simulador scrum sería:

Nada frecuente	Baja frecuencia	Medianamente frecuente	Altamente frecuente
----------------	-----------------	------------------------	---------------------

3. ¿Cuál es la probabilidad del 1 a 100 de que llegues a ser usuario del simulador scrum?

10	20	30	40	50	60	70	80	90	100
----	----	----	----	----	----	----	----	----	-----

d. Actitud hacia el uso

1. El uso un simulador de scrum en mi trabajo sería:

Muy bueno	Bueno	Normal	Malo	Muy malo
Muy beneficioso	Beneficioso	Normal	Muy dañino	Dañino
Muy sabio	Sabio	Normal	Absurdo	Muy Absurdo
Muy Positivo	Positivo	Normal	Muy negativo	Negativo
Muy placentero	Placentero	Normal	Aburrido	aburrido

e. Intención de uso

	Totalmente en desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	Totalmente de acuerdo
Tengo intención de utilizar el simulador scrum					
Me gustaría mostrar el sistema a mis conocidos					
Recomendaría el uso del simulador en mi trabajo					
Pienso que el simulador debería ser implementado en mi trabajo					

Autorizo el uso y la publicación de la información aquí plasmada con fines de investigación.

Nombre y Firma