



TESIS DE MAESTRIA

Modelación y aplicación de *Follow the Sun* para grupos de soporte ERP

por

Jonatan Caresani

Licenciado en Economía
2007 Universidad de Buenos Aires

Presentado a la Escuela de Posgrado del ITBA y de la EOI de España
en cumplimiento parcial
de los requerimientos para la obtención del título de

Magister en Dirección Estratégica y Tecnológica (Argentina)
Master Executive en Dirección Estratégica y Tecnológica (España)

En el Instituto Tecnológico de Buenos Aires

Agosto 2015

Firma del Autor _____
Instituto Tecnológico de Buenos Aires
Agosto 2015

Certificado por _____
Dr Alberto Terlato Profesor de Metodología de la Investigación
Instituto Tecnológico de Buenos Aires
Tutor de la Tesis

Aceptado por _____
MSc. Diego Luzuriaga Director del Programa
Instituto Tecnológico de Buenos Aires

Miembros del Jurado:

Agradecimientos

Al Dr. Alberto Terlato por sus buenos consejos

A mis empleadores y compañeros de trabajo por ayudarme tanto financieramente como prestando sus instalaciones y tiempo

Dedicatoria:

A mi familia y amigos por su paciencia y comprensión

A Diana Calderon Medellin por apoyarme constantemente

A Jorge Cham por su buen humor a cada momento

*I've got to get to you first
Before they do
It's just a question of time
Before they lay their hands on you
And make you just like the rest
I've got to get to you first
It's just a question of time*

Depeche Mode - A Question of Time

Índice Capítulos:

Abstract	Pág. VIII
Introducción	Pág. IX
Capítulo 1. Las metodologías FTS	
Alcance e inicios de FTS	Pág. 1
FTS y sus requerimientos	Pág. 3
FTS otra definiciones	Pág. 7
Framework para la modelización de FTS	Pág. 9
Tópicos de FTS	Pág. 12
Estudios empíricos	Pág. 14
Los desafíos presentes de FTS	Pág. 16
Capítulo 2. Los grupos de soporte ERP	
Introducción a los ERP	Pág. 20
Los grupos de soporte ERP en empresas globales	Pág. 24
Sería posible integrar FTS para ERP como metodología complementaria	Pág. 27
Capítulo 3. Desarrollo de un método de trabajo para grupos de soporte ERP aplicando metodología FTS	
Modelo FTS para grupos de soporte ERP	Pág. 34
Capítulo 4. Aplicación del método a un caso real	
Experimento de campo	Pág. 42
Diseño del experimento	Pág. 43
Resultados	Pág. 46
Capítulo 5. Conclusiones y propuestas	Pág. 50
Bibliografía	Pág. 52

Índice de Cuadros:

Cuadro 1. 1 Comparación de FTS respecto a otras Metodologías en eficiencia de calendario	Pág. 6
Cuadro 1. 2 Comparación entre las distintas definiciones de FTS	Pág. 8
Cuadro 1. 3 Los problemas más comunes para implementar FTS	Pág. 19
Cuadro 4. 1 Lista de participantes	Pág. 43
Cuadro 4. 2 Método de evaluación de la documentación	Pág. 44
Cuadro 4. 3 Checklist durante los hand off	Pág. 45
Cuadro 4. 4 Resultado de los hand off	Pág. 47
Cuadro 4. 5 Tiempo de resolución de los incidentes	Pág. 48
Cuadro 4. 6 Calidad de la documentación	Pág. 49

Índice de Figuras:

Figura 1. 1 FTS comparada con otras configuraciones de equipos globales	Pág. 4
Figura 1. 2 Actividades específicas de una fase que se amoldan a FTS de manera adecuada	Pág. 7
Figura 1. 3 Framework para modelización e investigación de FTS	Pág. 10
Figura 1. 4 Impactos de la falta de coordinación en la productividad	Pág. 17
Figura 2. 1 Departamentos integrados a un ERP	Pág. 22
Figura 2.2 Soporte ERP vía metodología Agile	Pág. 29
Figura 2.3 Escenarios apropiados para Agile	Pág. 29
Figura 2. 4 Ejemplo de Composite Persona	Pág. 31
Figura 3. 1 Los Estados de un CP durante su horario laboral	Pág. 35
Figura 3. 2 Modelo completo de FTS para soporte ERP	Pág. 39
Figura 4. 1 Distribución de los Equipos	Pág. 46

ABSTRACT

Follow the Sun (FTS) es una estrategia utilizada por empresas globales para el desarrollo de software utilizando las 24 horas del día, los siete días a la semana, a partir de grupos de trabajo distribuidos alrededor del mundo. Su utilización ha comenzado recientemente, tomando impulso a partir del año 2005, principalmente en países como Estados Unidos (Treinen, 2006).

Esto hace que los abordajes desde el mundo académico –tanto en sus aspectos técnicos como en lo referente a sus problemáticas – sean muy pocas y más que nada concentradas a su aplicación a proyectos de desarrollo de software.

Esta tesis intenta estudiar el alcance de dichas estrategias *FTS* para brindar soporte técnico en las áreas de *ERP (Enterprise Resource Management)* bajo la creencia de que la misma resulta suficiente para resolver los numerosos problemas que se presentan en la actualidad en dichas áreas y que obedecen a la falta de metodologías de abordaje de la gestión.

Finalmente este estudio intentará proponer un modelo para gestionar grupos de soporte *ERP* distribuidos en distintas zonas geográficas bajo la estrategia *FTS*.

Palabras claves:

Follow the Sun, FTS, Enterprise Resource Management, ERP, soporte técnico, estrategia, grupos de trabajo, Information Technology, globalización.

INTRODUCCION

Relevancia:

El Desarrollo Global del Software (*Global Software Development - GSD*) se ha vuelto una tendencia para las compañías de software que apuntan a mantenerse competitivas. *GSD*, o también conocido como Ingeniería Global de Software o Desarrollo de Software Distribuido puede ser definido como desarrollo de software con equipos localizados en distintos puntos geográficos. Se caracteriza por que uno o más de los integrantes del proyecto se encuentran físicamente distanciados uno del otro. Ello hace que los centros de desarrollo se hallen potencialmente en distintos usos horarios.

Follow The Sun (FTS) es una metodología de trabajo dentro de *GSD*. Comprende una idea relativamente simple: grupos de trabajo envían su labor todos los días a otro grupo para que puedan continuarlo a medida que el mundo va girando (de Francia a China por ejemplo) De esta manera mientras un grupo termina su turno otro continua con las tareas (Visser, 2009).

Con el avance tecnológico y el desarrollo de nuevas herramientas resulta cada vez más económico y efectivo poder lograr esta colaboración entre distintos grupos. Los beneficios que se esperan obtener adoptando esta metodología incluyen: *time-to-market* (reducir el tiempo de puesta en escena de un producto en el mercado), reducción de los costos de desarrollo y un mejor uso de los recursos escasos (Setamanit, 2006).

El impacto de aplicar *FTS* en el desarrollo de software puede ser profundo tanto a nivel teórico como práctico. En teoría una cantidad de grupos de trabajos puede organizar sus tareas de manera tal que se trabaje de manera secuencial diariamente gracias a la optimización de los costos de coordinación. El atractivo de esta metodología es poder reducir de manera sustancial el *time-to-market*.

A pesar de estos supuestos beneficios teóricos, como se verá más adelante, *FTS* ha tenido muy pocos casos de éxito en el desarrollo de software. Sin embargo se ha demostrado que

puede ser particularmente eficaz para reducir los tiempos en tareas que involucran reporte de errores y pruebas durante el desarrollo del software (Carmel, 2009). Esta conclusión es la que dispara la relevancia de esta investigación. Determinar si *FTS* puede ser aplicado para organizar tareas de una índole similar, adaptando su abordaje al soporte técnico de *ERP*. Este trabajo intentará estudiar si se puede utilizar el cuerpo teórico de *FTS* para resolver los problemas que encuentran los grupos de soporte de *ERP*. En lugar de descartar *FTS* por su poco éxito en el desarrollo de software intenta descubrir si es más adecuada para otras problemáticas.

Definición y alcance del problema:

FTS requiere una coordinación diaria de tareas entre grupos muy dificultosa. Esta restricción no ha cambiado drásticamente a pesar de los avances actuales en tecnología.. Estos problemas afectaron su desarrollo en implementaciones de software (Hess, 2012).

Como se verá un factor relevante que explica estas dificultades del trabajo bajo las metodologías *FTS* fue precisamente su principal ventaja, la diferencia en los usos horarios de cada grupo. Esto generó problemas de coordinación, a lo que se agregaron los factores culturales y las barreras de comunicación –que no siempre pueden resolverse con tecnología-. Desde lo tecnológico han surgido prácticas creativas como Agile, Cpro, entre otras, orientadas a reducir los costos de coordinación.

Al ser *FTS* una metodología nueva se requiere de una comprensión de los conceptos bases que son claves para una correcta implementación. Lo que lleva a uno de los puntos sobre los cuales estará apoyada esta investigación. ¿Es posible aplicar exitosamente dicha metodología a través de un modelo? ¿Puede éste obrar de guía para asignar tareas de una forma más óptima? ¿Puede un grupo de soporte *ERP* usarlo para trabajar dentro de la metodología *FTS*?

El objetivo de esta investigación no será demostrar, que las metodologías *FTS* en el desarrollo de software carezcan de utilidad. Sino que con una buena comprensión de la

metodología tal vez sea posible re utilizarla para otras áreas de *IT* como en el soporte de los *ERP*, imaginando que por la granularidad de sus tareas se vuelve un candidato ideal.

Estado del conocimiento e introducción a la metodología del abordaje:

Como se menciono anteriormente *FTS* es un campo relativamente nuevo por lo que el material publicado es escaso y casi nulo si se buscan aplicaciones de la metodología a otra área que no sean a nivel de desarrollo de software.

Para este trabajo fueron seleccionados un grupo de autores como referentes válidos. Otros, de carácter ampliatorio, fueron incluidos en la bibliografía consultada.

Se abordaron recientes *papers*, libros y artículos especializados. Entre ellos y sólo para citar algunos: el trabajo de Camel, Espinosa & Dubinsky (2010), *Follow the Sun Workflow in Global Software Development*, publicado en *Journal of Management Information Systems*, el *paper* de Kroll, Hashmi & Audy (2013), *A Systematic Literature Review of Best Practices and Challenges in Follow-the-Sun Software Development*, publicado en *Global Software Engineering Workshops (ICGSEW)*, el *paper* de Treinen & Miller-Frost (2006), *Following the Sun: Case Studies in Global Software Development*, el *paper* de Gupta, Mattarelli, Seshasai & Broschak (2009), *Use of collaborative technologies and knowledge sharing in co-located and distributed teams: Towards the 24-h knowledge factory*.

A partir de Treinen & Frost se presenta un análisis de las primeras aplicaciones de *FTS* en el desarrollo de software. Analiza tanto un caso de éxito como uno donde la implementación fue un fracaso. Del mismo desprende *take aways* donde resalta las problemáticas más comunes que se puede encontrar (barrera cultural y coordinación entre las más importantes). Sin demasiada rigurosidad enumera lo que se considera los diferenciales entre un caso y otro que permitieron el éxito de la aplicación de *FTS*. Camel, Espinosa & Dubinsky logran un cuerpo teórico mas acabado donde empiezan a dar forma a una definición más estricta *FTS* y cuáles son sus diferencias con otras metodologías. Los aportes de Kroll, Hashmi & Audy también son de vital importancia, en su trabajo hacen un

repasso de la bibliografía referida a *FTS*, no solo identifican similitudes sino también diferencia entre los disantos autores. Un trabajo esencial para entender el estado del conocimiento. Por último es importante recalcar los aportes de Gupta, si bien su trabajo no es netamente relacionado con *FTS* fundamente las bases para lo que es buenas prácticas para emprendimientos *IT* son equipos de trabajo alrededor del mundo. Sus sugerencias y conclusiones permiten deslumbrar que *FTS* también podría ser aplicado en otras aéreas si se consideran ciertos recaudos.

Hipótesis y objetivos de la investigación:

Las hipótesis que guían esta tesis han sido:

H.1: Es posible aplicar metodologías *FTS*, hasta hoy utilizadas en desarrollo de software, a grupos de soporte de *ERP*

H.2: Las metodologías *FTS* permitirían a los *ERP* (*Enterprise Resource Management*) resolver los numerosos problemas que presentan y que obedecen a la carencia de una metodología de gestión.

H.3: La aplicación de metodologías *FTS* a los *ERP* (*Enterprise Resource Management*) permite resolver incidentes de forma más eficiente, en tiempos menores.

Articulación:

En la Sección 1 se desarrollarán los aspectos clave de la metodología *FTS* para comprender su alcance. Se analizarán los aspectos de su aplicación y la enumeración de las problemáticas emergentes de su uso.

En la Sección 2, introducción a los grupos de Soporte *ERP*. Se describirá que hacen y cómo funcionan en las empresas globales. Asimismo se describirá que podría aportarle a un grupo de soporte de *ERP* una metodología *FTS*.

En la sección 3 se desarrollará un modelo para la gestión de grupos de soporte de *ERP* bajo la metodología *FTS*.

En la Sección 4, se desarrolla un experimento de campo con la aplicación del modelo presentado en el capítulo 3, se comenta su evaluación y métricas a considerar. Luego se discutirán los resultados obtenidos, estos permitirán sustentar la idea que este tipo de metodología es beneficiosa en el plano de soporte de *ERP* con respecto a otras metodologías de soporte (*around the clock*).

En la Sección 5, se confrontarán las hipótesis y se obtendrán las conclusiones. También se incluyen propuestas de futuras líneas de investigación.

CAPITULO 1. Las metodologías FTS

1. Alcance e inicios de FTS:

FTS es parte de las metodologías *GSD*. Es aplicado para poder tomar ventaja de la distancia temporal entre varios centros de operaciones localizados en diferentes zonas horarias (Visser, 2009). Su principal propósito está focalizado en reducir el ciclo de desarrollo del software o *time-to-market* (Carmel, 2011). No ofreciendo otra ventaja más que ello. Es aplicado en los proyectos de software cuando el producto necesita ser desarrollado de manera rápida y los costos no son tan importantes para el cliente. (Carmel, 2010)

Como los miembros del equipo están distribuidos en distintos usos horarios, las organizaciones pueden desarrollar software las 24 horas del día de manera continua. Así, la reducción del tiempo de desarrollo podría operar en magnitudes significativas hasta el 50% si se trabaja en dos centros y en un 67% si se combinan tres centros de desarrollo (Kroll, 2013).

Usando esta operatoria, al comienzo y al final de cada jornada laboral se produce un pase de las tareas. Esta entrega se la conoce en la literatura bajo el termino *hand off*, y comprende la transmisión de tareas de un centro a otro (Carmel & Espinosa, 2011). Los *hand off* se producen todos los días para presentar el *status* actual del proyecto y pasar las tareas sin terminar de un centro a otro. El próximo grupo siguiente trabajara en estas tareas al comienzo de día laboral (Visser & Solingen, 2009).

La práctica de *hand off* crea dependencia entre los centros. El equipo que comienza las tareas sin terminar depende del *status* que el anterior equipo le proveyó. En la literatura, *hand off management* es mencionado como uno de los principales desafíos al momento de implementar *FTS* (Carmel & Espinosa, 2011).

En la literatura, *FTS* también se lo conoce como *around-the-clock*. Aunque estos términos son utilizados en una manera similar, sus definiciones son distintas. *FTS* es sobre

velocidad, reducir la duración de un proyecto, mientras que *around-the-clock* es sobre cobertura las 24 horas, teniendo una operación activa durante todos los turnos (Carmel & Espinosa, 2011).

En proyectos del tipo *around-the-clock*, hay equipos con tareas específicas en cada centro. Cada equipo es responsable por cumplir determinadas tareas y no comparte tareas con el resto de los centros. Es la metodología más utilizada para brindar soporte técnico así como desarrollo de software actualmente. Grupos diseminados en el mundo dan soporte las 24 horas pero solo una persona determinada se encarga de la resolución de incidentes y no transfiere sus tareas a ningún grupo. Una sola persona tiene *ownership* del incidente (Carmel, 2006). Actualmente *FTS* es recomendado solo para las etapas de desarrollo y testeó en el desarrollo de software. Puede ser aplicado en otras fases, pero puede que no acorte los tiempos de esa fase o su implementación sea difícil. No tiene mención alguna para soporte técnico de ningún tipo (Carmel, Espinosa & Dubinsky, 2010).

FTS es atractivo para las empresas por su beneficio teórico de reducir los tiempos de desarrollo los cuales le permiten colocar en el mercado los productos de manera más rápida. Esto es importante para empresas que pertenecen a industrias donde los productos se vuelven obsoletos muy pronto. Un buen ejemplo son las aplicaciones para telefonía móvil. *Time-to-market* también es importante para sistemas de información estratégicos como pueden ser sistemas *e-commerce* o sistemas de gestión para *Supply Chain Management*, por nombrar algunos.

No obstante hoy en día esta sensación de urgencia es compartida por la mayoría de las firmas, los proyectos y mercados. Asimismo existen tensiones reactivas para reducir la duración del desarrollo de los proyectos trabajar menos horas extras, reducir costos, plantear fechas de entrega más agresivas o evitar agregar personal. No solo hacer lo que se debe hacer sino hacerlo más rápido para satisfacer al cliente en un mundo de servicios muy competitivo y comunicado.

Los primeros casos documentados de *FTS* datan de 2005 y fueron realizados por IBM en Estados Unidos (Treinen & Frost-Miller, 2006). Los mismos corresponden a desarrollo de software y concluyen que, para grupos con mismos niveles de experiencia y una cultura similar, se observa una leve mejoría en los tiempos pero que está lejos de los valores teóricos que se mencionaban anteriormente. Como desventaja observan que si las tareas se comparten con grupos con culturas y lenguajes radicalmente distintos (Estados Unidos e India) los costos de coordinación se vuelven muy grandes y los niveles de retrabajo se vuelven intolerables. En este último caso se optó por volver al formato *Around the Clock* y que cada grupo trabaje en una tarea particular.

2. *FTS* y sus requerimientos:

Según Carmel (Carmel, 2009) *FTS* implica satisfacer estos cuatro criterios, a saber:

1. Los centros de desarrollo se encuentran alejados y en distintos husos horarios.
2. Uno de los principales objetivos es la reducción de la duración del desarrollo.
3. En cualquier punto en el tiempo hay un solo sitio que está trabajando en el producto.
4. *Hand off* se practican diariamente, entendiendo como tal a la entrega del *status* del proyecto y el *check in* del próximo grupo para continuar trabajando en él.

Carmel, define el *FTS* como:

Un tipo de metodología de trabajo con equipos globales, diseñada para reducir la duración de los proyectos, en cual el conocimiento del producto pertenece y es desarrollado por un grupo de trabajo y es entregado de manera diaria al próximo grupo de trabajo que se encuentra en otra zona horaria para continuar el trabajo. (Carmel, 2009, p. 6)

Otros puntos importantes y supuestos sostenidos por Carmel (2009) para sustentar esta definición apuntan a considerar que:

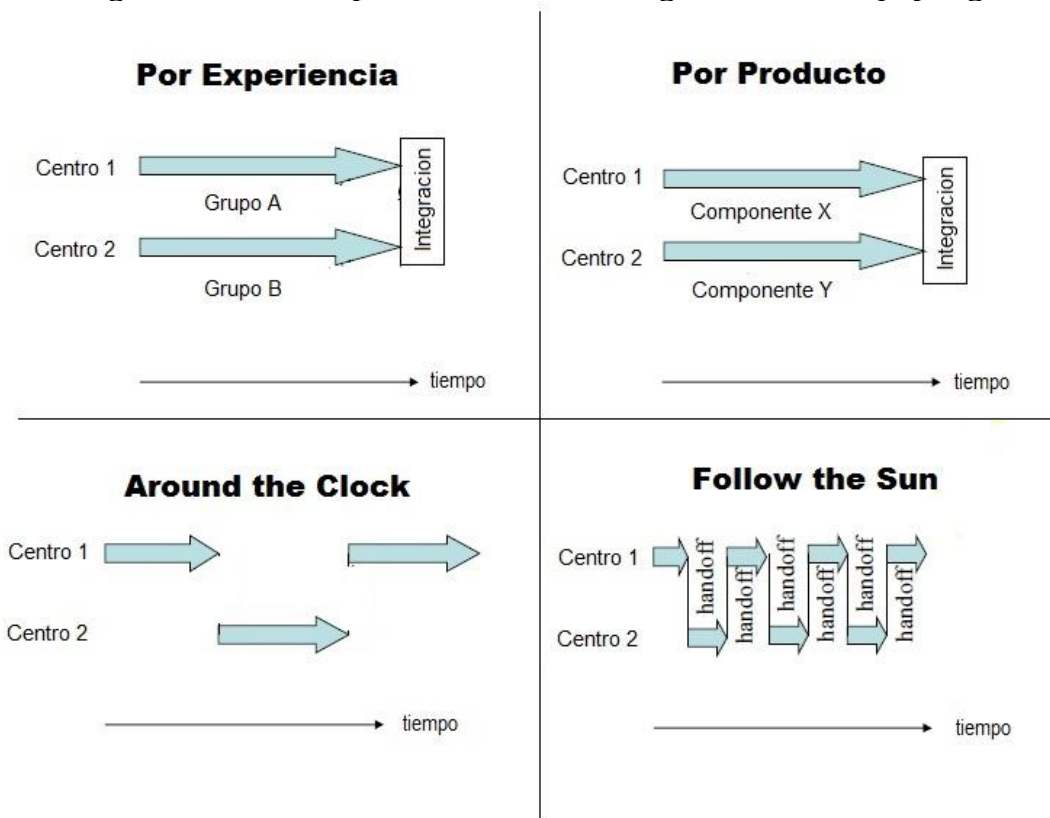
- Cada equipo trabaja solamente durante el día y descansa de noche.
- Cada equipo dentro de sí mismo trabaja como un equipo y coordina sus tareas.

- Cada equipo puede consistir de una o muchas personas.
- Existe un repositorio común donde todos los equipos realizan sus entregas de código o distintos tipos de avances.
- Se consideran excepciones donde los *hand off* no pueden ser realizados debido a fiestas, feriados nacionales o fines de semana.

Esta ampliación de su definición permite expandir la manera en que se puede utilizar *FTS* con 2 o 3 centros de soporte. Asumiendo 6 horas de intenso desarrollo de software o soporte técnico por día se podría considerar inclusive 4 centros diseminados en distintas zonas geográficas del globo.

El proceso de *hand off* de un centro a otro es clave para *FTS*. En la mayoría de los equipos globales se hace el mayor de los esfuerzos para hacer la menor cantidad de *hand off* posibles. En Figura 1.1 se pueden ver las configuraciones típicas para equipos globales

Figura 1.1 FTS comparada con otras configuraciones de equipos globales



Fuente: Carmel, Espinosa & Dubinsky, 2009

De la literatura referida a *GSD* se delinean tres conceptos similares, que nos son *FTS* (aplicables tanto para desarrollo como soporte):

- *Around the Clock*: esta etiqueta corresponde a los grupos de trabajos que colaboran y trabajan a través de múltiples zonas horarias. (Gupta, 2009) Aunque, en la mayoría de los casos estos grupos tienen muy poca dependencia de tareas entre sí y no realizan *hand off* para reducir los tiempos.
- Procesos de 24 horas: estos tipos de procesos son muy comunes en *call centers* ya que automáticamente redirigen las llamadas a los grupos que se encuentran activos en otra parte del mundo. Sin embargo, estos grupos tienen casi nula dependencia con otros grupos respecto a las tareas que realizan. Son utilizados para grupos de soporte que están más enfocados a la consulta y no tareas que involucren programación o configuración de un sistema. Si bien son ideales para soporte técnico por teléfono no se ajustan a las necesidades del soporte de un *ERP*.
- Locaciones con múltiples turnos: una alternativa razonable a *FTS* es elegir una locación donde los costos de mano de obra son bajos y tener turnos de 8 horas seguidos. Además de los bajos costos, los turnos pueden ser organizados de forma tal que se solapen para que tengan tiempo para poder transmitir cara a cara un *hand off* adecuado de las tareas. Este tipo de configuración si bien es viable no puede ser englobada como *FTS* ya que los grupos se encuentran en la misma locación. Es compatible tanto con desarrollo como con soporte, la mayoría de estos casos se observan en India (Carmel, 2006). Si bien es una alternativa atractiva del lado del costo del personal por los niveles de salarios en estos países sufre al estar asociada a un simple *outsourcing* de tareas de la compañía a una consultora. La compañía pierde control del proceso y se vuelve dependiente de la consultora (Saunders, Van Slyke & Vogel, 2004). Su estudio está fuera del alcance de esta tesis, ya que el objetivo se centra en utilizar *FTS* en empleados de la compañía y no en su proceso de tercerización.

En orden para comprender cuanto realmente impacta *FTS* en la velocidad de desarrollo así como en la resolución de tareas de soporte (la reducción de los tiempos en general es el tema central) Carmel (2009) introduce el término calendario de eficiencia, como un análisis del uso eficiente del tiempo calendario para la producción.

Dicho autor define como eficiencia del calendario al porcentaje que surge del cociente entre el tiempo utilizado disponible y el tiempo total calendario disponible ($24 \times 7 = 168$ horas semana). Así por ejemplo, una semana de 40 horas de trabajo utiliza el 23.8% del calendario para trabajo ($40/168$). En consecuencia, su eficiencia es solo de 23.8%, demostrando que hay mucho espacio para mejorar. Las empresas sometidas a necesidades de las entregas suelen apelar a utilizar más horas, por cierto más caras y de menor productividad.¹ En el cuadro 1.1 se puede observar diferentes combinaciones globales, incluyendo *FTS*, sin considerar mermas de eficiencia por sobrecarga horaria. De dicha tabla se puede observar que *FTS* es la alternativa que ofrece la mejor eficiencia de calendario y que está a su vez crece en la medida que aumenta el número de centros.

En referencia a la performance hay substancial evidencia en la industria que *FTS* puede ser efectivo para reducir los tiempos en fases específicas (Figura 1.2) como la etapa de test donde: por ejemplo un equipo buscar errores en el programa y los documenta en la base de datos, para que luego sea accedida por otro equipo en otro centro y zona horaria para trabajar en ellos.

Cuadro 1.1 Comparación de FTS respecto a otras metodologías en eficiencia de calendario

Centros/ Turnos	Descripción	Horas	Porcentaje	Máximas horas calendario por semana
1	Línea Base	40	23.8%	168
1	Línea base + 10 hs extras	50	29.8%	168
1	Línea base + 20 hs extras	60	35.7%	168
2	FTS	80	47.6%	168
2	FTS + 10 hs extras	100	59.5%	168
3	FTS	120	71.4%	168

Fuente: elaboración propia

¹ Cuando una persona trabaja más de 8 horas las horas que se agregan representan menor productividad por fatiga, afloran los problemas de salud y los conflictos domésticos los cuales a su vez impactan reduciendo la productividad en el trabajo. Igual para trabajos en los fines de semana

Nota: Supuestos y bases para los cálculos: Días por semana = 7, de los cuales laborales son solo 5; Horas por día = 8 horas por turno excepto en los casos donde se indiquen horas extras; Pérdidas o eficiencias en tiempo por coordinación son ignoradas en esta tabla de momento; Tiempo libre. Días por vacaciones o feriados son ignorados, ignorada improductividad laboral por sobrecarga laboral.

Figura 1.2. Actividades específicas de una fase que se amoldan a *FTS* de manera adecuada.



Fuente: Carmel, Espinosa & Dubinsky (2010)

Vale la pena mencionar que estas tareas mencionadas en la figura 1.2 son las más estructuradas y granulares para el proceso de *hand off*, un punto sobre el que se volverá en la sección 4.

3. *FTS* otras definiciones:

Independiente de lo previamente expuesto hay otras definiciones de *FTS* publicadas en la literatura. Gorton y Motwani (2008), por ejemplo, definen que *FTS* como relacionado con trabajos en turno para el desarrollo de software. Los autores muestran que el desarrollo de 24 horas puede ser logrado solo aprovechando las diferencias en los husos horarios de los distintos equipos. Cada equipo trabaja en su horarios regular y al final de la jornada la tarea es enviada al siguiente equipo, que está comenzando su turno y continua en donde quedo el trabajo pendiente por el anterior equipo. Treinen y Miller-Frost (2006) definen *FTS* de una manera muchas clara y concisa. Para estos autores, la diferencia horario debería ser vista como una ventaja para distribuir los equipos. La meta es crear un ambiente tal de desarrollo de software donde los equipos solo trabajan durante sus horas regulares de trabajo, y al final del día, solo reasigna las tareas que comienza a trabajar otro equipo, de esta manera generando efectivamente las 24 horas continuas de desarrollo. Visser (2009) define *FTS*

como una metodología para desarrollar software, que consta en tener equipos de desarrollo en distintas zonas geográficas. Cada equipo, al final del día, entrega información relevante respecto a que se realizó en el día, además entrega el código desarrollado al próximo equipo, que está comenzando su turno. El trabajo será continuado desde donde fue dejado. Así, el proyecto está siendo desarrollado en un lapso de 24 horas continuas y no solo por ocho horas, como es convencionalmente hecho.

Para Gupta (2009), los desarrollos con *FTS* pueden extenderse a diferentes tareas (no solo desarrollo de software). Carmel, Espinosa, y Dubinsky (2010), definen *FTS* como una metodología de trabajo global, que apunta a reducir la duración del proyecto, en cual el producto pertenece a un centro hasta que es entregado al próximo equipo. Estas entregas (*hand off*) deben hacerse de manera diaria y este equipo que recibe las tareas debe estar en una locación alejada tanto geográficamente como en husos horarios. Sin embargo, además de la financiación también pueden existir otros costos que surgen a la hora de evaluar un proyecto de inversión. En este trabajo se abordarán los tres más comunes: los costos hundidos, los costos de oportunidad y los costos por efectos colaterales.

Cuadro 1.2 Comparación entre las distintas definiciones de FTS

Referencia	Principal Objetivo de FTS			Hand Off Diarios		Considera relevante	
	Incrementar la velocidad de desarrollo	Reducir time-to-market	Incrementar el desarrollo realizado diariamente	Código de programa	Extenderse a otras tareas	Trabaja solo en horarios regulares	La diferencia horaria es por muchos husos horarios
Visser (2009)			X	X	X		X
Gorton y Motwani (2008)	X	X		X	X	X	
Gupta (2009)	X		X		X		
Treinen y Miller-Frost (2006)				X	X	X	
Carmel, Espinosa, y Dubinsky (2010)	X	X		X		X	X

Fuente: Kroll, Hess, Audy & Prikladnicki (2011)

Como se puede observar en el cuadro 1.2, no hay una total correspondencia entre las definiciones de *FTS*. Independiente de esto, todas se complementan entre sí, con características específicas, que contribuyen a definir mejor *FTS*.

- *FTS* es una estrategia global de desarrollo de software pero puede ser aplicada a otras áreas
- El objetivo principal es acelerar los tiempos de desarrollo del producto y en consecuencia reducir el *time-to-market*.
- Los equipos de trabajo se encuentran en distintos países y husos horarios
- Cada equipo trabaja en un determinado horario
- El proceso de *hand off* puede aplicarse a cualquier tarea
- El *hand off* debe ocurrir diariamente y de una manera estandarizada.

4. Framework para la modelización de *FTS*:

Entendiendo que es *FTS*, los problemas que enfrenta y cuáles son las maneras de enfrentarlos la construcción de un modelo aun necesita de ciertos lineamientos, un *framework* que asegure que al diagramarlo se esté respetando los conceptos claves de *FTS*.

La propuesta de Kroll, Silva Estacio, Audy y Prikladnicki (2014) comienza con la selección de al menos una variable independiente. Las variables independientes son el objeto de estudio ya que permiten ser asociadas a causas.

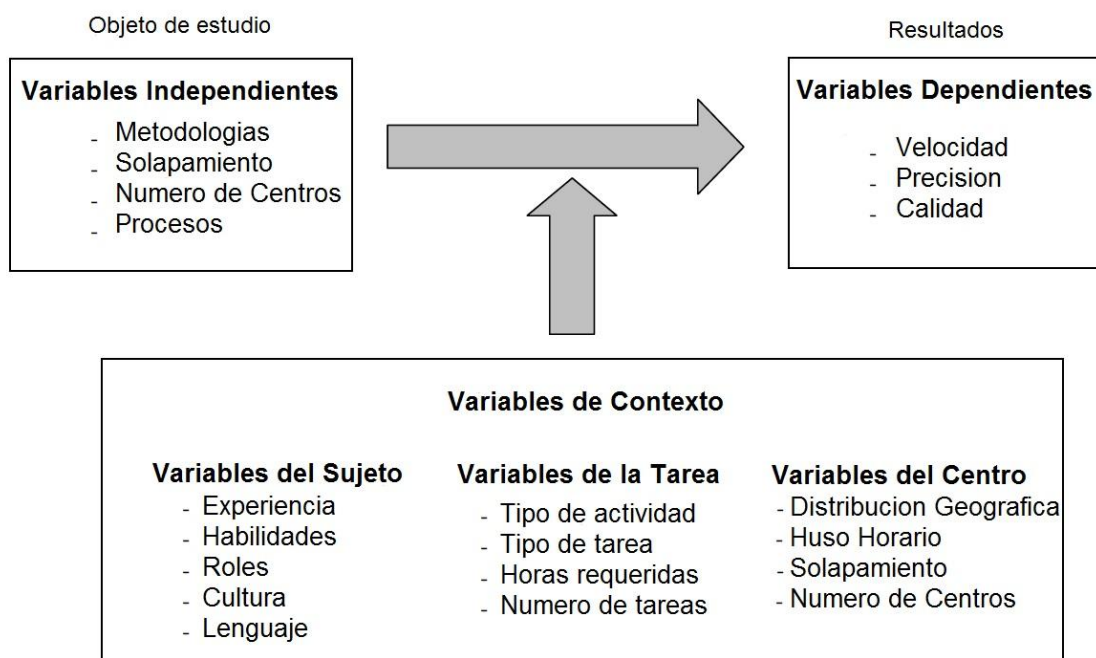
Las variables independientes son analizadas bajo un conjunto de variables de contexto seleccionadas para llevar a cabo el estudio. Las variables de contexto reflejan las condiciones en las cuales las variables independientes y dependientes son estudiadas.

Una variable dependiente es aquello que se pretende medir de un estudio, y que es afectado durante el mismo. Las variables dependientes responden a las variables independientes al menos existan otras variables no identificadas o el modelo tenga un comportamiento

azaroso. Las variables independientes proveen capacidad explicativa pueden servir para medir en que grado explican la variable explicada (la dependiente) por tanto sirven para contrastar las hipótesis, poder afirmarla o refutarla.

Para tener una mejor comprensión de la relación entre las diferentes variables del *framework* seleccionado se ha construido la Figura 1.3.

Figura 1.3 Framework para modelización e investigación de FTS



Fuente: Kroll, Silva Estacio, Audy y Prikladnicki (2014)

Variables independientes:

- Metodología: es importante porque pueden colaborar con el desarrollo de *FTS* y alcanzar un nivel de productividad óptimo. Una de las más comentadas en la literatura y que se menciona es *Agile* (Yap, 2005).
- Solapamiento: en *FTS*, los miembros del equipo colaboran con otros miembros que están en otras zonas horarias. La existencia o no de solapamiento en los turnos es clave para un

buen *hand off*. Cuanto mayor sea el solapamiento mayores son las probabilidades de una mejor coordinación y comunicación entre los centros.

- Número de centros: cantidad de éstos, a medida que aumenta el número de centros las probabilidades de terminar una mayor cantidad de tareas es mayor, pero la coordinación y comunicación entre los centros será más compleja. Se presenta un *trade off* clave para el éxito de la implementación de *FTS*.
- Procesos: un proceso es utilizado para designar una serie de herramientas, métodos y prácticas para completar una tarea. La definición de un proceso tiene gran impacto sobre la productividad. Esto se refleja en el conjunto de buenas prácticas sugeridas por distintos autores (Kroll, Hashmi, Richardson & Audy, 2013)

Variables de contexto:

- Variables del sujeto: son las características que definen el sujeto de estudio, son muy importantes para determinar el éxito del proceso.
- Variables de la tarea: pueden afectar los resultados de *FTS*. En este caso de incluyen tipo de actividad, horas requeridas y numero de tareas.
- Variables del centro: distribución geográfica, huso horario, solapamiento y número de centros son variables importantes que afectan la performance del grupo. Un manager de proyecto tiene estas variables en cuenta al gestionar un proyecto de software o *IT*.

Variables dependientes:

- Velocidad: es la variable dependiente más importante para *FTS*. Tiene una relación directa con cómo se mide la eficiencia ya que afecta los ciclos de vida de desarrollo y los tiempos que toma poner el producto en el mercado. También demuestra la eficiencia en que se cumplen las tareas.

- Precisión: se considera precisión el grado en que el trabajo realizado se parece al esperado (Kroll, Hashmi, Richardson & Audy, 2013). En *FTS* al pasar una tarea de un centro a otro centro se espera que las especificaciones sean claro y no lleven a malentendidos que se reflejan en tener que rehacer las tareas.
- Calidad: es un factor importante para el éxito de un proyecto (Hess & Audy, 2012). La calidad tiene una relación fuerte con la precisión. Generalmente se mide en función del trabajo enviado y cuanto fue terminado satisfactoriamente.

5. Tópicos de FTS:

Como se menciono previamente si bien el concepto de *FTS* se ha visto prometedor en teoría, su aplicación ha resultado compleja. Mientras más detalle conlleve el proceso de traspaso de tareas, menor va a ser el nivel de de ambigüedad librado al azar, de esta manera las probabilidades de que ciertas tareas deban ser rehechas será menores también. Por otro lado, el solapamiento entre los sitios debería ser lo menor posible para dejar la mayor cantidad de tiempo posible dedicada al desarrollo de software o tareas pertinentes.

En el campo de las colaboraciones a través del mundo, las opiniones respecto al largo de los *hand off* varían. Schwaber (2009) recomienda reuniones diarias de 15 minutos, mientras que otros sostienen que reuniones semanales de entre 30 o 60 minutos son suficientes como para mantener a todos los miembros al día (Treinen & Miller-Frost, 2006). Por otro lado, hay compañías que tratan de maximizar el solapamiento durante las horas de oficina, cambiando las horas laborales en una región en otra zona horaria (Carmel, 2006) o se planifican llamadas por fuera del horario laboral para otras regiones (Gupta, 2009).

Para ver los efectos que tienen la diferencia horaria, Espinosa y Camel (2003) construyeron un modelo de costos de coordinación. La primera versión de este modelo consta de dos sujetos donde los costos son la variable dependiente. La variable costo consiste de

producción, coordinación y costos de vulnerabilidad². Espinosa y Carmel concluyen que los costos decrecen cuando las horas de solapamiento aumentan, lo que se podría concluir es que un solapamiento total es preferido. Sin embargo, un aumento en la diferencia horaria disminuye los efectos de un aumento en la coordinación y aumenta la duración de las tareas. En 2004, extendieron el modelo para poder considerar equipos más grandes y poder evaluar los efectos de las diferencias horarias en los costos de coordinación. (Espinosa & Carmel, 2004). De esta versión extendida Camel y Espinosa concluyen que menos diferencia horaria reduce los costos de coordinación pero aumenta los costos de vulnerabilidad. Incrementar las horas de solapamiento reduce ambos costos, por lo que más solapamiento siempre es preferido.

Según los principios de *FTS*, existe la obligación a comunicarse con otros centros al menos dos veces al día. Primero con el centro que le transfiere el trabajo y luego al final del día con el centro que va a recibir los avances en el proyecto. Esto implica que el solapamiento no debería ser tan largo, porque la comunicación es frecuente. Es cierto, el solapamiento no debería ser largo, porque *FTS* hace hincapié en entregar el trabajo a otro grupo y no en trabajar en conjunto con otro grupo en otra locación al mismo tiempo.

Esta idea se pone en manifiesto en los tres niveles de relaciones temporales, identificado por Kumar, Van Fenema and Von Glinow (2005). Ellos definen tres tipos de colaboraciones: *clean hand off*, donde la interacción es corta, *sticky hand off*, donde la interacción es más intensa y trabajo en conjunto, cooperación completa entre centros durante todo el día. Tanto *clean* como *sticky hand off* pueden funcionar con *FTS* mientras que trabajo en conjunto no. Siguiendo las conclusiones de Treinen y Miller-Frost (2006) en su caso de estudio de IBM, *sticky hand off* son los más adecuados.

Por último un tema que se trata en distintos trabajos es los factores que afectan la comunicación entre centros. Si bien la comunicación en tiempo real o no es la más tratada

²Son los costos que se generan por una mala comunicación o demoras en la entregas porque los miembros de un equipo están distanciados geográficamente. Los costos de vulnerabilidad representan el tiempo que toma rehacer una tarea o identificar los malentendidos que se produjeron en una comunicación y el tiempo que toma corregirlos.

los efectos de distintas lenguas en cada centro son innegables. Las compañías optan por un idioma común a todos los centros (Visser & Solingen, 2009), que es inglés en la mayoría de los casos (Kroll, Hess, Audy & Prikladnicki, 2011) lo que puede convertirse en una barrera para poder expresarse libremente para algunos miembros del equipo. Una práctica recomendada es la inversión en cursos de inglés para los que no tiene el idioma como lengua madre.

6. Estudios empíricos:

Kroll (2013) identificó estudios cuando condujo una revisión sistemática de la literatura a fin de conocer las mejores prácticas para la implementación de *FTS*.

Camel (1999) reportó el primer experimento utilizando *FTS* en la industria del software. En este caso, IBM creó cinco equipos en cinco centros distintos localizados en cinco países. En este proyecto, IBM enfrentó muchos problemas de coordinación, específicamente durante los *hand off* diarios. Como *FTS* no estaba generando los resultados esperados y la cantidad de problemas se acrecentaban, los responsables del proyecto decidieron abandonar *FTS* y optaron por continuar con tareas distribuidas entre distintos grupos en diferentes zonas geográficas donde ninguna de ellas interactuaba.

Yap (2005) describe la experiencia de usar *XP (Extreme Programming)* en un proyecto con *FTS*. En este estudio, el autor reporta desafíos, lecciones aprendidas y soluciones para la integración de equipos globales como diferencias culturales y conflictos emergentes entre centros. Este estudio se centra particularmente en *XP* aplicado particularmente a *FTS* como estrategia de desarrollo de software. Luego de estar utilizando *XP* con grupos en distintas locaciones a lo largo de un año la autora concluye que es posible trabajar con grupos distribuidos en distintos países. Si los programadores comparten un repositorio de datos común y ponen especial atención a los *hand off* es posible. Algunas de las ventajas que se resaltan es la posibilidad de acortar los tiempos de desarrollo, contacto en tiempo real con los clientes en distintas regiones y mayor capacidad para adaptarse a un cambio. También se reconoce que hay muchas aéreas que deben seguir siendo estudiadas pues hay espacio para introducir mejoras.

Treinen y Miller-Frost (2006) dan detalles sobre dos casos de estudio en IBM. El primero describe un proyecto de desarrollo de software entre USA y Australia. En éste, dos centros geográficamente distanciados fueron fusionados en un solo equipo para adaptarse a una estrategia *FTS*. El caso fue considerado un éxito. En el segundo caso se estudian tres proyectos distintos, con centros en USA e India. En éstos, los equipos trabajaban sobre el mismo código. Debido a la inexperiencia del equipo, limitaciones de tiempo, presupuesto del proyecto el mismo fue cancelado. Se encontraron muchos desafíos debido a diferencias de horario, diferentes configuraciones en el ambiente de desarrollo, mala estimación de costos y amplias diferencias culturales entre los equipos.

Espinosa, Nan y, Carmel (2007) condujeron el primer experimento controlado de *FTS* para investigar el efecto de solapamiento de los husos horarios sobre la velocidad y la precisión. En este estudio, 42 pares de estudiantes fueron elegidos para simular un proyecto *FTS*. Este estudio demostró que la velocidad de desarrollo era más alta tanto para total solapamiento como para el nulo, pero no en diferentes fracciones de solapamiento entre los sitios. Se usaron mapas ficticios para evaluar el impacto del solapamiento sobre la velocidad y la precisión.

Carmel, Dubinsky y Espinosa (2009) realizaron un cuasi experimento para medir la velocidad de desarrollo en *FTS* entre los equipos. Este estudio siguió las prácticas de la metodología Agile. Los resultados fueron que los equipos utilizando *FTS* fueron más rápidos que equipos solo con metodología *around-the-clock*. En este estudio, el tiempo de desarrollo se incrementó un 10%, pero, para los autores, podría ser más alto aun.

Kroll (2012) investigó los beneficios de un acercamiento adaptativo y prescriptivo para *FTS*. Este estudio también utiliza mapas ficticios. Los resultados obtenidos han indicado que el uso de métodos adaptativos incrementan la velocidad, pero no siempre implican una mejora en la calidad y precisión del trabajo realizado por los centros.

Hess y Audy (2012) proponen un proceso para lidiar con los *hand off* diarios. Un experimento controlado fue realizado para evaluar su eficiencia. La evidencia muestra que es posible reducir las dificultades de implementar *FTS* utilizando este proceso. El proceso está basado en *CP (Composite Persona)*.

Kroll (2013) realizó un estudio de caso sobre *Infosys Technologies* para examinar la viabilidad y los resultados de *FTS*. Por un mes, equipos de desarrollo en México, India y Australia trabajaron en el desarrollo de aplicaciones utilizando *FTS*. Este estudio presenta detalles sobre las prácticas de software y las soluciones aplicadas para superar los desafíos encontrados.

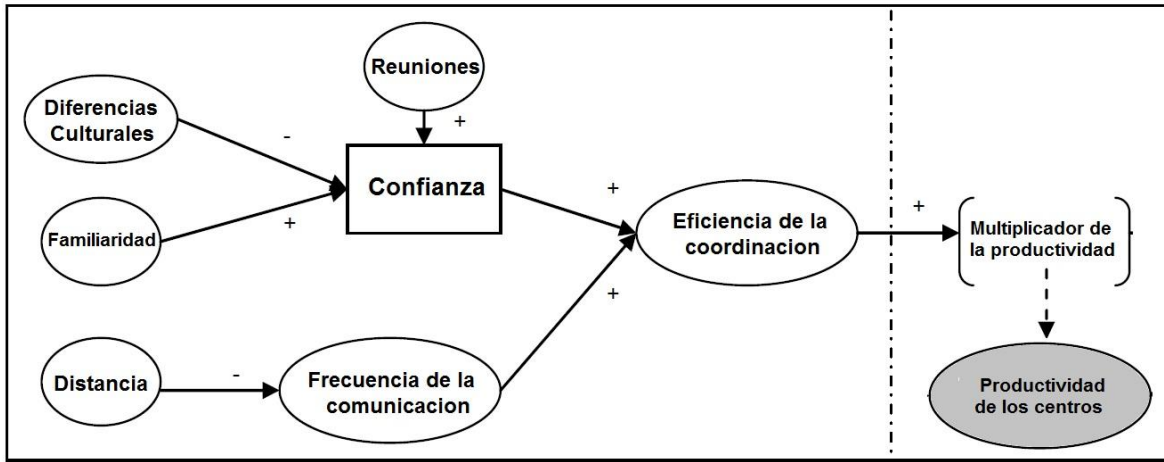
Kroll, Richardson & Audy (2014) investigaron como debería ser tratado los procesos de *hand off* en un contexto de *FTS*. Los resultados describen la percepción de los participantes sobre las actividades realizadas, los desafíos enfrentados y sus respectivas soluciones.

7. Los desafíos presentes de FTS

Hay problemas que enfrentan la mayoría de las implementaciones de *FTS* que se debe tener en cuenta si se pretende una implementación: evitar las fallas de coordinación, mejorar la comunicación y resolver las diferencias culturales.

Los problemas de coordinación derivan en pérdidas de productividad (Solingen & Valkema, 2010). Debido a la distancia geográfica y las diferencias horarias entre los centros, la intra e inter coordinación entre los centros es más difícil, lo que amplifica aún más los impactos en la productividad.

Figura 1.4 Impactos de la falta de coordinación en la productividad.



Fuente: Setamanit, Wakeland & Raffo (2006)

Hay dos factores primarios que tiene un impacto significativo en la eficiencia de la coordinación: la frecuencia de la comunicación y confianza. Los equipos coordinan su trabajo a través de comunicaciones para intercambiar información (avances, *status* del proyecto, etc.) Distintos informes sobre desarrollo de software (Carmel, 2006) demostraron que la comunicación informal, no planificada y *ad hoc* es extremadamente importante para la colaboración entre equipos. La discusión entre pares es la más usada y valorada de las técnicas de coordinación (Gupta, 2009). De esta manera, equipos que se comunican frecuentemente entre ellos tienden a coordinarse mejor. Sin embargo, la distancia entre miembros de equipo afecta negativamente los niveles de comunicación (Carmel, Espinosa & Dubinsky, 2010).

La confianza es un valor clave en cualquier equipo. Sin confianza, es poco probable que un equipo trabaje de manera conjunta efectivamente ya que los miembros del equipo no están dispuestos a comunicarse abiertamente. Equipos con altos niveles de confianza tienden a coordinar mejor y en consecuencia logran mejores resultados (Kroll & Audy, 2012).

La confianza (Terlato, 2014) es un valor que se puede crear o destruir desde el liderazgo y desde el sistema de gestión. Los sistemas de gestión muy enfocados en el monitoreo y control (enfoques de gerencia de tipo X según la clasificación realizada por McGregor)

tienden a volver a las personas poco comprometidas con su trabajo, su compromiso llega hasta realizar lo requerido, los niveles de cooperación son bajos y bajos los niveles de cooperación y confianza intergrupales. Para desarrollar mayores niveles de adhesión, compromiso y cooperación, asimismo confianza y mejorar las comunicaciones se necesitan desarrollar sistemas de gerencia diferentes (a los que McGregor denomina de tipo Y) donde las personas se hallan empoderadas, desarrollan iniciativas, crecen en autoconfianza, mejoran su autocontrol y en donde los jefes se parecen más a un *coacher* que a monitores del principio de la revolución industrial.

Personas de la misma cultura desarrollan confianza de manera más rápida precisamente porque la comunicación es mejor que entre personas de distintas culturas. Personas de una cultura más individualista tienden a estar propensos a confiar en otros que las personas provenientes de una cultura donde se coopera más intensamente (Lee, DeLone & Espinosa, 2008). La confianza aumentará a medida que los miembros del equipo se vuelvan más familiares el uno con el otro. Desafortunadamente, sin comunicación cara a cara, la confianza tiende a decaer en el tiempo (Santos, Sales & Fernandes, 2012). Precisamente porque detrás de la incomunicación está el miedo. El miedo eclipsa la comunicación. Es posible mejorar la comunicación y la confianza modificando el sistema de gestión y reduciendo los niveles de desconfianza.

Los costos generados por la coordinación son la vértebra de porque es tan difícil de aplicar *FTS*. La coordinación, es por definición, el trabajo necesario para administrar las dependencias entre las tareas desarrolladas por varios miembros de un equipo (Espinosa, Cummings, Wilson & Pearce, 2003). Ahora no solo deben realizar sus tareas pero además apartar tiempo para coordinar e integrar su trabajo. Para que *FTS* logre cumplir su promesa de acotar los tiempos, los costos de coordinación deben ser minimizados.

En la literatura (Carmel, Espinosa & Dubinsky, 2010) se descompone en tres componentes los costos de coordinación:

- La coordinación en el centro

- La coordinación entre centros (actividades de *hand off*)
- La productividad personal

La coordinación entre centros está relacionada con el concepto de eficiencia de *hand off*. El trabajo terminado junto con las tareas pendientes que se envían requieren de una presentación acorde para que puedan ser entendidos. Si no es apropiadamente preparado puede prestarse a confusión y generar que muchas partes tengan que ser re hechas nuevamente. Algunos factores que afectan la eficiencia del *hand off*:

- Fallas del *hand off*: cuando una de las partes falla en entender lo que le fue enviado por el centro anterior.
- Tiempo variable en el *hand off*: a medida que aumentan la cantidad de centros se acumula mucha más información que debe ser compartido con los otros centros
- Tiempo dedicado al *hand off*: en la medida que se dedique poco tiempo al *hand off* las probabilidades de generar malentendidos son mayores

Para detallar más explícitamente algunos de los obstáculos que se da por sentado, se agrupa los problemas asociados a FTS en el Cuadro 1.3

Cuadro 1.3 Los problemas más comunes para implementar FTS

Dificultad	Referencia
Coordinación	
Diferencias Horarias	Carmel & Espinosa (2011) (2003) Yap (2005)
Procesos de <i>hand off</i> diarios	Hess & Audy (2013) Carmel (2006)
Dispersión geográfica	Carmel, Dubinsky & Espinosa (2009)
Falta de cohesión como equipo	Carmel, Dubinsky & Espinosa (2009) Espinosa & Carmel (2003)
Dificultades técnicas	Yap (2005)
Comunicación	
Diversidad socio-cultural	Hess & Audy (2012) Carmel & Espinosa (2003)
Comunicación en tiempo real	(Gupta, 2009) (Treinen & Miller-Frost, 2006)
Distintos idiomas	(Carmel & Espinosa; 2003) (Tang & Inkpen; 2001)
Feriados y fiestas nacionales	(Kroll, Santos, Prikladnicki & Hess, 2012)
Cultural	
Diferencias culturales	(Visser; 2009) (Kroll, Hess Audy & Prikladnicki 2011)
Distintos niveles de experiencia	(Yap, 2005) (Treinen & Miller-Frost, 2006)

Fuente: Kroll, Hashmi, Richardson & Audy (2013)

CAPITULO 2. Los grupos de soporte ERP

1. Introducción a los *ERP*:

Enterprise Resource Planning (ERP) es un *software* para el control de gestión — usualmente conformado por un conjunto de aplicaciones integradas—que una compañía puede utilizar para recolectar, almacenar, analizar e interpretar información de distintas áreas de la empresa, entre ellas:

- Planeamiento de la producción y costos
- Manufactura y entrega
- Marketing y ventas
- Manejo de inventario
- Pagos y cobros

ERP permite tener una visión integradora de los núcleos principales de los procesos que componen una empresa, generalmente en tiempo real, utilizando bases de datos comunes.

Los sistemas *ERP* monitorean los recursos de la compañía — efectivo, materias primas, capacidad productiva—y el estado de los compromisos adquiridos: órdenes de venta, compras, y liquidación de sueldos por poner un ejemplo. Las aplicaciones que componen un *ERP* comparten información entre los distintos departamentos (producción, compras, ventas, contaduría, etc.) que proveen la información (SAP Web Oficial). Los *ERP* facilitan el flujo de la información entre las distintas funciones del negocio y administra la conexión de esta con los *stakeholders* ajenos a la compañía (Hosseini, 2004).

ERP es una industria de miles de millones de dólares que produce componentes para soportar las más variadas necesidades de los negocios que comercializan distintas compañías alrededor del mundo como Oracle (JDE), SAP (SAP), Sage (Sage Group), Microsoft Dynamics (Microsoft) o Infor (Infor Global Solution). Desde el lado de la demanda, es

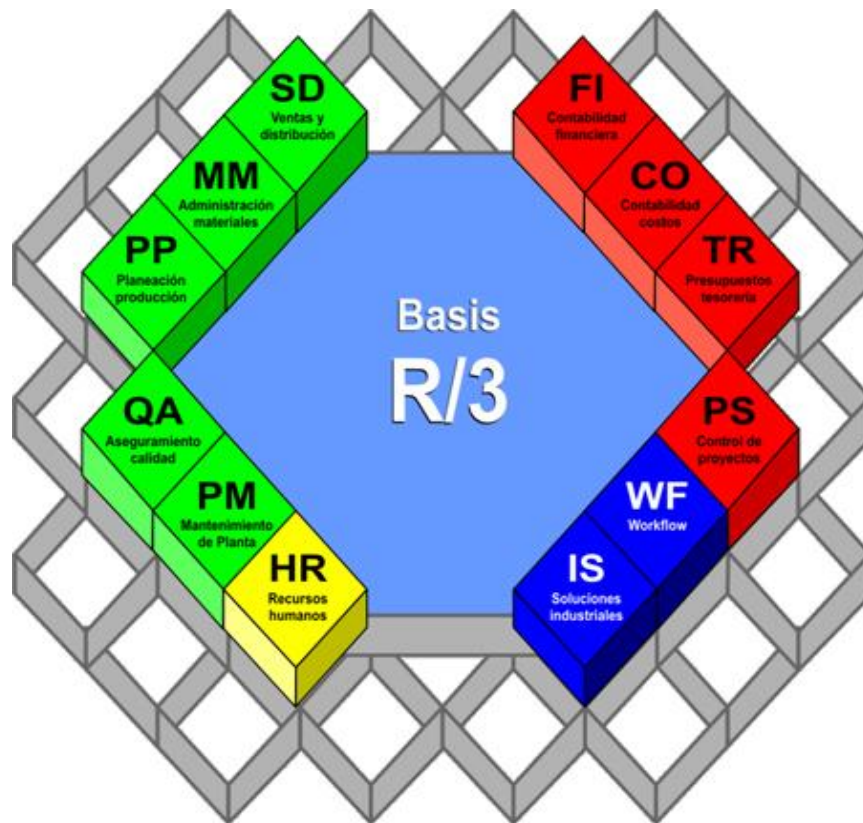
decir desde las empresas que utilizan este tipo de aplicaciones, las inversiones en *IT* se han vuelto una de las categorías de mayor foco. Estudios realizados en la última década en los Estados Unidos la ubican en el primer lugar dentro de las inversiones de capital. Si bien en los primeros años los *ERP* se focalizaban en grandes empresas, hoy también las pequeñas y medianas, de manera incremental, están adoptando estos sistemas (Rubina & Kotze, 2011).

Los *ERP* son considerados una herramienta vital en las organizaciones porque integran muchos sistemas y facilitan transacciones libres de error al validar las distintas áreas de la compañía. El desarrollo de este *software* es distinto del de sistemas tradicionales (Shaul & Tauber, 2012). Los *ERP* corren en una variedad de hardware mayor y la configuración de su red es más compleja, en la mayoría de los casos usando la base de datos como un repositorio de información (Khosrow-Poor & Mehdi, 2006).

En 1990, *Gartner Group* fue el primero en utilizar las siglas *ERP* como una extensión de *MRP* (*Material Requirements Planning*), al integrar las ciencias de la computación e *IT* en el proceso de manufacturación. Así *ERP* pasó a significar la evolución de aplicaciones integradores que van más allá del proceso de manufactura. (Anderegg, 2013)

No todos los paquetes de *ERP* vienen de la integración entre *IT* y la manufactura. Algunos comenzaron soportando los procesos Contaduría o Recursos Humanos. Para mediados de los 90 la mayoría de los departamentos de una compañía tenían su contraparte en *ERP*. Gobiernos y organizaciones sin fines de lucro también empezaron a implementar sistemas *ERP*. Fue durante el año 2000 con el problema de Y2K y la introducción del euro como moneda que varios sistemas *legacy* se volvieron obsoletos, muchas compañías tomaron esto como una oportunidad para reemplazar sus viejos sistemas con un *ERP*. (Sheilds, 2005)

Figura 2.1 Departamentos integrados a un ERP



Fuente: Web oficial de SAP (www.sap.com)

Los sistemas *ERP* inicialmente estaban focalizados en automatizar tareas de *back office* que no afectaran directamente al cliente y al público general. Luego comenzaron a abarcar otras aéreas como las de manejo de clientes (*CRM*) o integrarse a sistemas *e-business* como *e-commerce*, *e-government*, *e-telecom*, y *e-finance*. A partir de allí se integraron al manejo de proveedores (*SRM*), gracias al avance de la Internet que simplifico la comunicación con terceros (Shaul & Tauber, 2013).

El término "*ERP II*" fue acuñado en 2000 en un artículo por *Gartner Publications* titulado Los ERP están muertos – Larga vida a *ERP II*. Describe como el software basado en navegadores provee acceso en tiempo real a los real tanto a empleados como terceros (proveedores o clientes). *ERP II* expande el rol de solo optimizar recursos y procesar transacciones. *ERP II* maneja la información disponible para colaborar con otras empresas (Anderegg, 2013). *ERP II* es más flexible que la primera generación de *ERP*. No concentra

las capacidades del sistema dentro de la compañía, va más allá para interactuar con otros sistemas.

Los *ERP* ofrecen una serie de ventajas sobre otros software que los vuelven muy atractivos para las compañías (Walsh, 2009)

- Predicción de ventas en base a historial, lo que permite optimización del inventario.
- Historial cronológico de cada transacción con información relevante de las áreas involucradas en la operación.
- Monitoreo de órdenes y de ganancias por periodo o área.
- Comparación entre lo comprado, lo recibido y los costos asociados.
- Elimina la necesidad de tener que sincronizar cambios entre múltiples sistemas (consolidación de finanzas, ventas, marketing recursos humanos y producción).
- Aporta legitimidad y transparencia a los datos estadísticos.
- Promueve una vista general de la compañía (no son islas de datos), generando información en tiempo real disponible para la toma de decisiones.
- Protege datos sensible al consolidar múltiples sistemas de seguridad en una sola estructura
- Mejora la eficiencia de una compañía al integrar los departamentos de la misma.

En la actualidad los desarrolladores y proveedores de *ERP* están haciendo un esfuerzo para integrar periféricos móviles (*smartphones* por ejemplo) a los sistemas *ERP* con nuevas aplicaciones. Los desafíos desde el aspecto técnico incluyen crear nuevo *hardware*, aplicaciones, redes y agregar cadenas de valor. Otra consideración importante es que los *ERP* cubren mas funciones y roles que nunca, por lo que un foco en la estandarización de procesos y transparencia de los mismos es vital (Shaul & Tauber, 2013).

2. Los grupos de soporte *ERP* en empresas globales:

Un grupo de soporte *IT* es necesario para asegurar el buen funcionamiento de un *ERP* así como para atender las mejoras que este requiera. En la mayoría de los casos por una cuestión de economías de escalas el soporte de un *ERP* es tercerizado a consultoras, solo empresas globales con centros dedicados al soporte técnico, poseen empleados bajo su nombre. También se da el caso donde conviven consultores externos y empleados de la empresa (Carmel & Tjia, 2005)

Al ser un sistema embebido en las necesidades del negocio un perfil netamente técnico muestra ser inadecuado. Para cubrir el soporte se requieren dos tipos de perfiles diferentes: analistas funcionales y técnicos.

El analista funcional es el nexo entre el negocio y *IT*, conoce del funcionamiento del sistema desde el apartado funcional (lo que debe hacer) y las necesidades del negocio. Se encarga de la configuración del sistema, monitoreo del mismo y atender consultas del negocio. Al ser tan amplia las áreas cubiertas por un *ERP* los analistas funcionales se especializan según el departamento, que, en la jerga, es conocido como modulo. Los tipos de analistas funcionales más comunes son:

- Finanzas: contaduría, activos fijos, cuentas a pagar, cuentas a cobrar, tesorería, manejo de caja, presupuestos y costos.
- Recursos humanos: reclutamiento, entrenamiento, liquidación de sueldos, beneficios, políticas de diversidad y ética, retiro y terminación del empleado.
- Producción y manejo de materiales: compra de materiales, control de stock, planeamiento de compras y producción, control de calidad, proceso de manufactura, ciclo de vida del producto.
- Ventas: órdenes de venta, revisión de crédito, precios, inventario, envíos, facturamiento, análisis de ventas y reporte, comisión por ventas.
- *Supply chain management*: planeamiento de *Supply chain*, agenda de proveedores, alineación con el proceso de ventas, compras, inventario, proceso de reclamos, manejo de almacenes (recepción, registro, *picking* y embalado).

- Administración de proyectos: planeamiento de proyectos, planeamiento de recursos, costeo, WBS, facturamiento por horas trabajadas y gastos, métricas de performance.
- Atención al cliente (*CRM*): ventas y marketing, comisiones, servicios, centros de atención al cliente, integración a *call center*.

Por otro lado está el perfil meramente técnico, quién trabaja con el analista funcional para que el sistema pueda adaptarse a las necesidades del negocio, su gestión subdivide en distintas tareas, los perfiles más comunes son:

- Diseño y mantenimiento de bases de datos
- *Hardware* y redes
- Performance del sistema (back ups, memoria y sistema operativo)
- Coordinación de transporte de cambios
- Programadores

El alcance que tiene un *ERP* implica cambios significantes en los procesos y rutinas diarias de los empleados (Sheilds, 2005). Generalmente los tiempos de resolución de un equipo de soporte varían en función de la complejidad del proceso, el número de módulos requeridos, la envergadura del problema y los tiempos que se consideran aceptables para resolver el problema. Un *upgrade* del sistema general para una empresa grande puede tomar alrededor de 14 meses y requiere de alrededor de 150 consultores. Los proyectos pequeños pueden requerir meses y si se consideran cambios en empresas multinacionales e implementaciones de mayor o igual escala pueden tomar años. Si la necesidad de soluciones específicas es necesaria los tiempos de implementación pueden ser aun mayores (Bingi, Sharma & Godla, 1999).

La mayoría de los grupos de soporte *ERP* como los *ERP* en sí mismos incorporan mejores prácticas. Esto significa que tanto el soporte como el *software* reflejan la interpretación de los prestadores de las maneras más efectivas de realizar cada tarea. El soporte como el *ERP* varía en función de que tan convenientes sean para el cliente. Las compañías que

implementan las mejores prácticas reducen tareas que consumen tiempo como pueden ser configuración, documentación, testeo o entrenamientos.

Las practicas (*IFRS*, *Sarbanes-Oxley*, o *Baselia III*) reducen la cantidad de defectos en alrededor un 71% comparado con otras implementaciones de *software*. *IFRS* (*International Financial Reporting Standards*) es un lenguaje común para llevar el registro contable de una compañía. De esta manera los libros contables de distintas compañías en distintas partes del mundo son comparables entre sí. *Sarbanes–Oxley* es una ley federal de los Estados Unidos, determina los estándares para todas las compañías públicas de este país. *Baselia III* son recomendaciones para leyes que regulen el sector financiero y son emitidas por El Comité de Basilea. Los *ERP* deben considerar estas normas al momento de su diseño ya que son las prácticas comunes alrededor del mundo para también para ser comparables entre sí. (Sheilds, 2005).

El soporte de *ERP* tiene un amplio rango para mejorar. Un estudio de Monk, & Wagner (2009) identifica como una de las principales fallas, los niveles de desconocimiento del funcionamiento del *ERP* de las empresas clientes debido a los altos niveles de *outsourcing* de los grupos de soporte. Un punto sobre el que se volverá en el capítulo 5 con las conclusiones.

Este mismo estudio, sostiene que las consultoras pasan a tener control completo del conocimiento del sistema, así la empresa pierde control del *know how* o posibilidad alguna de crear un *knowledge transfer* y así se vuelve dependiente de los servicios de la consultora. Otra de las críticas gira alrededor de los tiempos de resolución elevados debido a situaciones conflictivas entre los usuarios de la empresa y los grupos de soporte tercerizados (Monk, & Wagner, 2009).

Desde una perspectiva técnica se agregan algunos problemas desde el área de soporte (Young, 2014):

- Las soluciones específicas para una empresa pueden ser problemáticas para otra. Los *ERP* tratan de cumplir las necesidades bases de la mayoría de las compañías, forzando a la organización a encontrar alternativas dentro del sistema para suplir esas demandas.
- La reingeniería de los procesos de negocios para poder adaptarse a las necesidades del sistema *ERP* pueden dañar la competitividad o desviar el foco de otras actividades críticas.
- Tanto el soporte como el *ERP* pueden costar más que soluciones menos integradoras o no tan abarcativas.
- La dependencia del sistema *ERP* puede aumentar el poder de negociación del proveedor de software y soporte. Reflejándose en un aumento de los costos de soporte, mantenimiento y mejoras.
- Largas sesiones de *training* consumen mucho del tiempo de empleados que deberían estar focalizándose en sus tareas operativas diarias.
- Debido a la arquitectura de los *ERP* (*OLTP*, procesamiento de transacciones en tiempo real) puede que las soluciones para producción y planeamiento como *SCM* se tornen más complejas que otras soluciones de *software*.
- La armonización del sistema *ERP* pueda ser una tarea colosal (especialmente para compañías grandes). Requiere una cantidad de tiempo, planeamiento y dinero importante.

Este trabajo considera que, dado que el núcleo del costo del soporte proviene desde las cantidades de horas de consultoría pactadas, las empresas globales deberían apoyarse en nuevas metodologías para poder reducir estas horas. La metodología *FTS* proveería no solo una reducción de los tiempos sino una mejor diversificación del *know how* dentro del grupo.

3. Sería posible integrar *FTS* para *ERP* como metodología complementaria:

Carmel, Dubinsky y Espinosa (2009) proponen un acercamiento a través de metodologías *Agile* (desarrollo ágil de software). Esto se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios (Hazzan & Dubinsky, 2008). Existen muchos métodos de desarrollo ágil; la mayoría

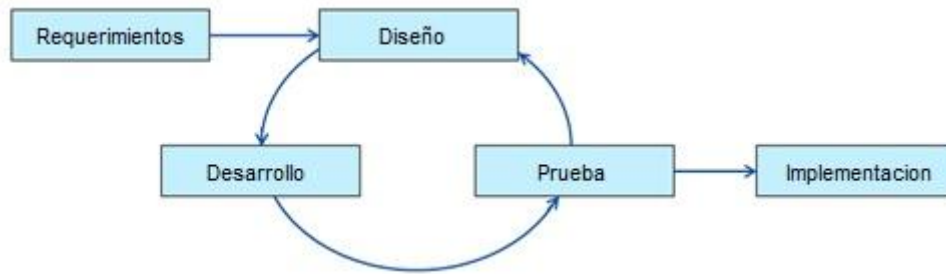
minimiza riesgos utilizando lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración o ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiado detalle de funcionalidad para justificar el desarrollo, sino que posee como meta lograr una «demo» (sin errores) al final de la iteración, luego de la cual el equipo vuelve a evaluar las prioridades del proyecto (Sutherland, Viktorov, Blount & Puntikov, 2007).

Agile es una filosofía que usa modelos organizacionales basados en personas, colaboración y valores compartidos (Sutherland, Viktorov, Blount & Puntikov, 2007). Sus valores son:

- Atención en las personas y sus relaciones
- Compromiso con el trabajo en equipo
- Priorizar la entrega de una solución sobre documentación exhaustiva
- Equipo auto organizado
- Planeamiento constante y priorización basado en las necesidades del cliente y la performance del equipo
- Flexibilidad
- Recursos con experiencia

Bajo estos valores hay una premisa donde tanto el usuario como el grupo de soporte trabajan en equipo y se prioriza la solución por sobre la burocracia. La resolución de un problema en el sistema se vuelve más dinámica mediante constantes iteraciones.

Figura 2.2 Soporte ERP vía metodología Agile



Fuente: Elaboración propia

Se debe tener en cuenta que *Agile* solo puede ser aplicado en contextos donde el balance entre la tecnología a soportar y los requerimientos sean medianamente conocidos. Si fueran completamente conocidas las iteraciones se vuelven innecesarias ya que claramente se sabe qué hacer. Un planeamiento detallado sería más oportuno para este tipo de escenarios (Hazzan, O. and Dubinsky, Y., 2008)

Figura 2.3 Escenarios apropiados para Agile



Fuente: Hazzan, O. y Dubinsky, Y. (2008)

Agile aprovecha y resalta las fortalezas del equipo (Yap, 2005):

- *Accountability*
- Empoderamiento

- Comunicación y colaboración
- Focalizados en la visión y no en una tarea en particular

Esta metodología posee características que podrían ayudar a estructurar un modelo *FTS*:

- Soporte a *hand off* diarios: continuamente integrando los avances de un equipo con el otro. Los valores centro de *Agile* inculcan una disciplina que empuja a los miembros del equipo a continuamente interactuar entre ellos. El proceso de *hand off* se volvería natural para el equipo.
- Comunicación: herramientas y mecanismos automáticos que dan soporte a los aspectos técnicos, liderazgo y sociales que puede presentar el trabajo. Dejando de lado la necesidad de mantener canales de comunicación todo el tiempo. Para un claro y eficiente traspaso de tareas se necesita de una estructura que lo permita. Las recomendación de *Agile* respecto a cómo llevar una reunión, organizar el trabajo y otorgar *accountability* de las tareas permiten que los grados de dependencia entre los individuos del equipo sean menores.
- Colaboración: formalización de protocolos que permiten al próximo centro seguir trabajando con niveles mínimos de comunicación entre centros. Aumenta la cantidad de tiempo que los miembros del equipo pueden dedicarle al soporte de *ERP*, en lugar de estar advocating a tareas administrativas pueden trabajar de lleno en los problemas que presenta el *ERP*.

Agile provee un sistema iterativo que es así ideal para *FTS*, compila el trabajo de un ciclo y provee las herramientas necesarias para apalea los problemas de comunicación que una operación de este tipo puede encontrar. La necesidad de *hand off* diarios obliga a llevar un nivel de detalle de las tareas que las mismas se vuelven granulares y se pueden re acomodar en tareas más sencillas que son mucho más maniobrables. Ergo más adaptadas a *FTS*.

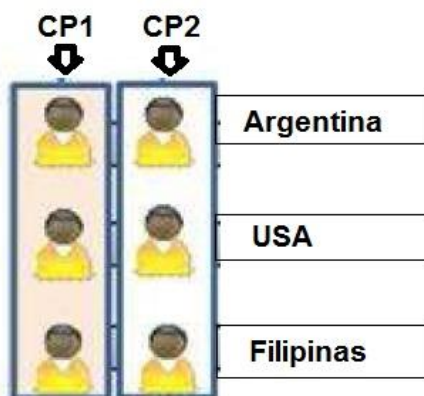
Una vez que el trabajo ha sido descompuestas en tareas más simples, inclusive es posible asignar las tareas al centro que tenga mayor experiencia en ese tema (Carmel, 1999), independiente de la locación o huso horario. Si bien esto puede ser considerado como incompatible con la idea de *FTS* se debe tener en cuenta que no todos los modelos de *FTS* son netamente puros, no todo el trabajo tiene que requerir de *hand off* necesariamente. También se puede pensar que no es necesario llevar el trabajo a este nivel de granularidad, solo el necesario para que funcione *FTS*.

Anteriormente se puso de manifiesto la importancia de los *hand off* en *FTS*, pues pueden detener la producción si las especificaciones no son claras. Para realizar los *hand off* diarios entre centros se utilizan un sinnúmero de herramientas: llamadas por teléfono, mensajería *on line* o sistemas de tickets documentando los avances y hallazgos.

Para lograr una mejor cohesión entre los miembros de los equipos y mejorar la documentación la utilización de *CP* (*Composite Persona*) resultará una herramienta útil. *CP* consiste agrupar personas de distintas locaciones y grupos en un subgrupo, se les asignan tareas y objetivos en común propios de este subgrupo. Así un conjunto de individuos de distintas locaciones pasan a formar un nuevo equipo virtual con tareas en común.

Para mitigar el problema anterior de retardo de comunicación asíncrona, agravada por los problemas de las diferencias de idioma y cultura, se desarrolló el concepto *Composite Persona* (*CP*); un *CP* aparece como un solo individuo con el mundo exterior, pero es en realidad compuesto por tres personas, con una persona en cada uno de los sitios que colaboran.

Figura 2.4. Ejemplo de Composite Persona.



Fuente: Elaboración propia

Cada *CP* está formado por al menos un miembro de un centro, como se ve en la figura 2.4, así se forman subgrupos entre los centros. Se ven obligados a trabajar en equipo y por lo tanto hacer un mayor esfuerzo por entregar la información de la manera más clara y concisa

posible. Así las tareas ya no pertenecen a un centro o una persona sino al conjunto de centros.

Este escenario de trabajo distribuido atenúa tanto la separación espacial como temporal mediante la involucración de personas en distintos países y explotando la diferencia de horaria como una ventaja, al permitir que las tareas se trabajen de manera secuencial (Gupta, 2006). Uno de los principales obstáculos es utilizar una técnica que permita transmitir y entender los avances de manera rápida. Para mitigar este problema el concepto de *Composite Persona (CP)* es ideal. Cada *CP* está encargado de distintas tareas, para adentro de cada *CP* hay una comunicación constante entre los miembros. Cada uno sabe que para un buen funcionamiento y resolución de los objetivos en común deben cooperar entre ellos. Esta constante interacción entre las partes produce una mayor cohesión entre los equipos de distintas locaciones. Como se ha mencionado anteriormente la confianza es un valor núcleo si se desea que una operación, que involucra distintos países, tenga éxito. *CP* provee una metodología de trabajo compatible con *FTS* al brindar soporte durante las 24 horas. Identifica un grupo que se mantiene en el tiempo, ayudando a inculcar los *hand off* en los miembros del equipo. Al ser un equipo que se mantiene en el tiempo saben que deben cumplir con sus obligaciones de traspaso de información.

El proceso de *hand off* se reduce a tres preguntas con *CP*

- i. ¿Qué has hecho desde la última reunión?
- ii. ¿Qué estás planeando hacer hasta el próximo encuentro?
- iii. ¿Existe algún impedimento o bloqueador?

Después de completar esta información, el trabajo se considera entregado al siguiente miembro del equipo virtual. El siguiente sitio comienza su jornada de trabajo recogiendo la información proporcionada por el sitio anterior y la definición de lo que debe hacerse, utilizando como referencia principal, las respuestas a las preguntas i, ii y iii. Este trabajo pone de manifiesto la importancia de contar con el personal equivalente en todos los sitios

distribuidos. Esta equivalencia no está relacionada con el número de recursos en cada sitio, pero en las capacidades de solución de problemas.

En resumen, *CP* para *FTS* provee ventajas (Denny, Mani, Sheshu, Swaminathan & Samdal, 2008) en los siguientes aspectos: mejores *hand off*, incrementa la confianza entre los miembros de los centros, acelera la toma de decisiones al estar centrada en menos individuos y genera acceso a 24 horas a soporte.

CAPITULO 3. Desarrollo de un método de trabajo para grupos de soporte *ERP* aplicando metodología *FTS*

1. Modelo *FTS* para grupos de soporte *ERP*:

Como en todo modelo, se requieren apelar a simplificaciones de la realidad para poder explicar un fenómeno complejo. A continuación se explicitan los supuestos bajo los que se construye este modelo:

- No hay trabajo basado en fases. Los equipos de trabajo *Agile* trabajan en todas las tareas.
- El personal es intercambiable. Si se lo piensa no es tan descabellado como uno podría pensar, muchas operaciones *IT* con centros en distintos países tienen un nivel de experiencia similar. Cabe aclarar que el personal que es intercambiado será reemplazo por otro de su misma especialidad, así por ejemplo un funcional de finanzas solo puede ser reemplazo por un funcional de finanzas así como un programador es reemplazado por otro programador.
- En caso de ausentismo se rompería el traspaso de tareas al próximo centro. Se considera que se da un *clean hand off* al más próximo vía correo electrónico.
- No hay trabajo en paralelo, solo secuencial.
- En caso de fallas en el proceso de *hand off*, el trabajo puede ser reprocesado por el próximo centro.
- El tiempo que toma realizar un *hand off* es fijo. Con el paso del tiempo para la resolución de incidente³ la información que se acumula es mayor. Suponemos que se alcanza un límite tal que el tiempo que toma el *hand off* es fijo.
- El tiempo que toma procesar la información recibida en el *hand off* es igual al tiempo que toma transmitirla.
- Las tareas realizadas son igualmente complejas para todos los centros y todos los miembros de cada centro están dedicados a estas tareas.

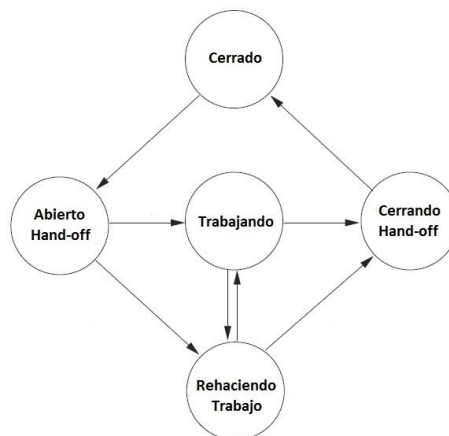
³ Defecto encontrado en el sistema o una consulta que se quiera hacer sobre el mismo. Para una mejor administración de su resolución hay herramientas que permiten llevar un *log* del progreso del mismo. En este caso se utiliza la herramienta Remedy.

- Los centros pueden transferir sus tareas tanto de manera eficiente como ineficiente. Por ineficiente nos referimos es que los *hand off* pueden estar incompletos o poco claros generando costos de vulnerabilidad.
- Cada centro esta subdividido en *CP* que se relacionan con otros centros. Cada participante de los *CP* trabaja en un conjunto de tareas, prepara el traspaso de las mismas y las transfiere al próximo centro.
- La construcción de este modelo está basado en 3 centros con turnos de 9 horas, donde se solapan entre ellos en la última hora y la primera del próximo centro.
- Los *hand off* se realizan en vivo vía conferencia.

En referencia a cada miembro del *CP*, como puede verse en la Figura 3.1, puede estar afectado por cinco estados por los que puede pasar durante el transcurso del día laboral:

- Cerrado: cuando el horario en la zona geográfica en la que se encuentra, la persona está fuera de horario laboral
- Abierto – *Hand off*: al comienzo del día recibiendo las tareas de otro miembro de su *CP*.
- Trabajando: continua con sus tareas diarias avanzando en la resolución.
- Rehaciendo trabajo: parte de las tareas realizadas por el centro anterior tienen errores y deben ser corregidas.
- Cerrando – *Hand Off*: pase de las tareas al próximo miembro del *CP* en otro centro.

Figura 3.1. Los Estados de un *CP* durante su horario laboral



Fuente: Elaboración propia

Una vez modelizado el comportamiento de cada integrante del *CP* es necesario adecuarlo en un contexto con procesos y reglas que permitan relacionarlo con los objetivos del centro y sus pares.

Asimismo correlacionado con las mejores prácticas comentadas en la literatura (Kroll, Hashmi, Richardson & Audy, 2013). Estas fueron sintetizadas en seis subprocesos que garantizan el mejor comportamiento de un modelo *FTS*: Formación de Equipos, Definición de Procesos, Protocolo de Comunicación, Training de Diversidad Cultural, Asignación de Recursos y *Hand Off*. Su correlación con las buenas prácticas (BP) es la siguiente:

Formación de Equipos:

BP01- Solapamiento de los turnos: generar una ventana de tiempo para que los centros se puedan comunicar en tiempo real.

BP02- El solapamiento se da al final de un turno y al comienzo del turno del otro centro.

BP03 – En lo posible formar equipos en distintos centros que compartan el idioma.

BP04 – Al menos una hora de solapamiento entre los centros.

BP05 – Los equipos deben estar distribuidos en dos o tres centros máximos, más aumentarían los problemas de coordinación.

BP06 – Niveles de experiencia similares entre los equipos para que la calidad de trabajo entre los centros sea equilibrada.

Protocolo de Comunicación:

BP07 – Acceso a medios de comunicación para que los centros puedan estar en contacto.

BP08 – Sistema de incidentes para informar sobre los avances en el mismo.

BP09 – Incitar la comunicación cara a cara en lo posible. Se recomienda durante las reuniones compartir pantalla para una demostración más efectiva durante las reuniones.

BP10 – Emails después de hora: para dar notificación en caso de emergencia para evitar que el trabajo se detenga.

BP11 – Comunicación informal y *ad hoc*: un nivel de rigurosidad más bajo es necesario para una mayor camaradería entre los centros.

BP12 – Modelos de emails y formularios: describen la información requerida según el tipo de mensaje a enviar.

Training de Diversidad Cultural:

BP13 – Reuniones de equipo para construir confianza.

BP14 – Formar equipos que compartan la misma cultura, tienden a generar vínculos de confianza más rápido.

BP15 – Entrenamiento para entender las distintas culturas y la riqueza de la diversidad.

Definición de Procesos:

BP16- Método *Agile* para adaptarse rápidamente a los requerimientos.

BP17- Procesos formales de documentación: reglas claras para documentar avances y cambios.

BP18- Repositorios de datos donde se compartan e intercambien documentos.

BP19- Equipos de *back up* en caso de urgencias para proveer un servicio de cobertura 24/7

BP20 – Sistemas para poder seguir el progreso del equipo.

BP21 – Administrar las horas laborales de cada equipo para que sean en horas de oficina normales y se logre el solapamiento con otro centro.

Asignación de Recursos:

BP22- Grupos en clave *CP*.

BP23 – Reducir el trabajo a un conjunto de tareas para una mejor comunicación de las mismas entre centros.

BP24 – Distribución de las tareas por secuencia o dependencia de manera que nos aseguremos alguien está trabajando en ella constantemente.

Hand Off:

BP25 – Intercambio diario del progreso del trabajo en reuniones de 30 minutos.

BP26 – *Clean & sticky hand off*. Los primeros apuntan a preguntas concisas: que se logró hasta el momento, que se piensa lograr hoy y cuáles son los obstáculos para cumplirlo. Los *hand off* pegajosos son interacciones más intensas pero son efectivas también.

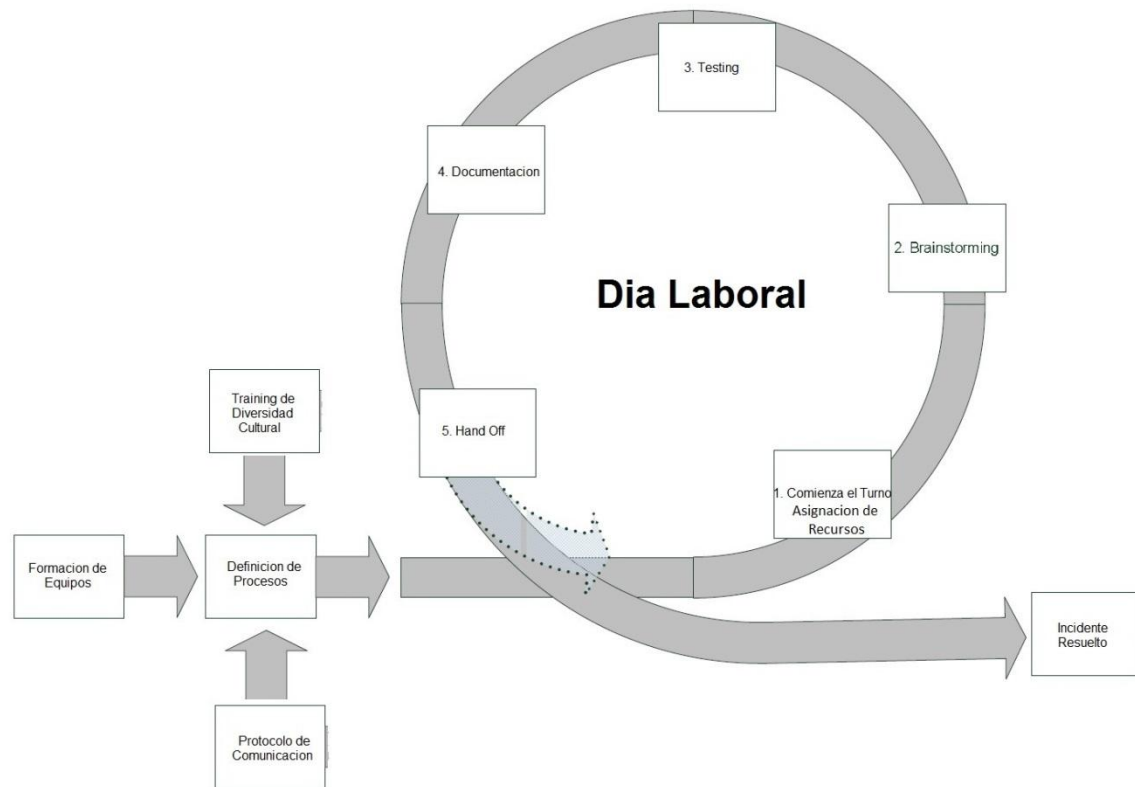
BP27- Formulario de *hand off* para controlar la calidad del mismo.

La formación de equipos da comienzo al proceso. Identifica los lugares donde hay recursos para crear un centro y contrata los perfiles adecuados para la tarea. Definición de Procesos, Protocolo de Comunicación, Training de Diversidad Cultural comienzan a correr en paralelo una vez que los centros y los recursos están disponibles. El protocolo de comunicación define con qué medios y de qué modo los miembros de los equipos entraran en contacto. Los training de diversidad cultural establecerán un mejor entendimiento de los distintos centros y forjara mayor confianza. Si es necesario con el paso del tiempo estos training se pueden reforzar.

Al comienzo de cada día se produce la asignación de los recursos, las tareas son asignadas a cada *CP* en función de su área de especialización. Así cada *CP* cubre durante las 24 horas un tipo de tema en particular. Cuando está por terminar el día laboral del centro, se da pie al proceso de *Hand Off*. El objetivo es transmitir lo realizado hasta la fecha y verificar que lo transferido ha sido comprendido por el próximo integrante del *CP*. El proceso termina cuando la tarea es terminada y el usuario confirma que la solución provista es satisfactoria. De lo contrario se repite la iteración (ver figura 3.2)

Así los subprocesos devenidos de las buenas prácticas y metodologías complementarios se integran al día laboral en el centro

Figura 3.2 Modelo completo de FTS para soporte ERP



Fuente: Elaboración propia

El resto de los pasos a seguir explican los pasos que cada miembro del equipo realiza diariamente

1. Comienzo del Turno:

Este estado representa el principio del día laboral para un centro. Se asignan los incidentes que se encuentren reportados por los usuarios y se acepta el *hand off* del centro previamente activo.

2. Brainstorming:

Una vez revisada la información provista por el miembro de su *CP*, la persona que se encuentra en este turno comienza a planear las tareas que debe realizar. Se continua

buscando información sobre el problema, entender la funcionalidad y su relación con otras áreas. Propone soluciones o nuevas ramas por donde llevar la investigación.

3. *Testing*:

Las soluciones propuestas deben ser probadas en lo posible en un ambiente de *testing*. Los resultados deben compartirse con el resto de los miembros del *CP*. Si la solución propuesta resuelve el problema es presentada al usuario y se implementa. En caso que necesite de asistencia a nivel técnica o funcional deberá interactuar con otros equipos durante su horario.

4. Documentación:

Todos los cambios realizados en el sistema sean de configuración o programación deben ser especificados detalladamente en los documentos pertinentes. Además los avances generales en el incidente deben ser reportados en el sistema de defectos o tickets que se utilice. Tanto el usuario como el equipo de soporte están informados respecto a los avances en el incidente.

5. *Hand Off*:

Hacia el final del día, cada miembro de un *CP* debe hacer los preparativos para el proceso de *hand off*. Preparar un informe a modo de presentación donde se informan los avances en los incidentes trabajados, básicamente debe responder cuatro preguntas:

- i. ¿Que fue realizado durante el turno?
- ii. ¿Cómo debería continuar el trabajo?
- iii. ¿Qué dificultades se encontraron en el camino?
- iv. ¿Que se espera obtener durante el próximo turno?
- v. ¿Es claro lo previamente expuesto?

Este estado representa el final del día laboral. Para este momento, una nueva persona toma el incidente y continua trabajando en él, en el caso que fuera resuelto este es cerrado y se continua con el resto de los incidentes o bien se asignan nuevos.

CAPITULO 4. Aplicación del método a un caso real

1. Experimento de campo

Para reducir la complejidad y el alcance del problema se diseñó un experimento focalizado la investigación en un solo modulo funcional en particular del ERP. Las tareas a realizar fueron específicamente seleccionadas para que el nivel de dificultad sea previsto antes de comenzar así como homogéneo para los distintos participantes.

Cuando se está utilizando *FTS*, la disciplina es crítica. El experimento de campo debería mantener un delicado balance entre mantener una atmosfera profesional y amigable entre los miembros de los equipos y mantenerlos al tanto que una falla en transferir los avances en tiempo y manera puede tener serias implicaciones para el experimento. Es claro que, al trabajar con gente en distintos husos horarios, completar las tareas a tiempo sería crucial.

Para el experimento las variables dependientes que se pretendieron medir fueron calidad y la velocidad. El experimento debería permitir confrontar las tres hipótesis planteadas, demostrar:

H.1: Es posible aplicar metodologías *FTS*, hasta hoy utilizadas en desarrollo de software, a grupos de soporte de *ERP*

H.2: Las metodologías *FTS* permitirían a los *ERP* (*Enterprise Resource Management*) resolver los numerosos problemas que presentan y que obedecen a la carencia de una metodología de gestión.

H.3: La aplicación de metodologías *FTS* a los *ERP* (*Enterprise Resource Management*) permite resolver incidentes de forma más eficiente, en tiempos menores.

Para el experimento se compararon los resultados entre un grupo de soporte de *ERP* que siguió los lineamos de *FTS*, presentados en el modelo descripto en el capítulo 3, contra un grupo que lo hizo trabajando con una metodología *around the clock* (*ATC*, soporte 24 horas) cuyas particularidades fueron descriptas en el punto 2 del capítulo 1.

2. Diseño del experimento:

Para llevarlo a cabo se pudo contar con el apoyo de una empresa que generosamente cedió el tiempo de sus empleados así como activos (servidores y sistemas) para reorganizar el modo en que se operó a lo largo de una semana (entre el día 27 de abril y el día 1 de mayo de 2015).

El grupo en cuestión ha estado conformado por consultores que se encargan habitualmente de dar soporte a un sistema SAP en el área de ventas y distribución. Los participantes resuelven tanto consultas sobre el sistema como así fallas en el funcionamiento del mismo a través de incidentes.

Participantes: los participantes del experimento desarrollan diariamente tareas de soporte. En promedio tiene más de 4 años de experiencia en el mercado. Por pedido expreso de la empresa tanto su nombre como el de los empleados serán reservados en confidencialidad.

Cuadro 4.1. Lista de participantes

Sujeto	Posición	Estudios	Locación	Experiencia	Equipo
S1	Consultor Senior	Ing Industrial	Argentina	8 años	FTS
S2	Consultor Jr	Lic en Sistemas	USA	1 año	FTS
S3	Consultor Semi Senior	Contador	Filipinas	4 años	FTS
S4	Consultor Semi Senior	Lic. en Adm Empresas	Argentina	5 años	ATC 1
S5	Consultor Senior	Ing. en Sistemas	USA	6 años	ATC 1
S6	Consultor Jr	Lic. en Sistemas	Filipinas	2 años	ATC 1
S7	Consultor Semi Senior	Sociólogo	Argentina	4 años	ATC 2
S8	Consultor Senior	Ing. en Informática	USA	7 años	ATC 2
S9	Consultor Jr	Contador	Filipinas	1 año	ATC 2

Fuente: Elaboración propia

Tareas a realizar: los participantes resolvieron incidentes planteados por los usuarios finales para el modulo de distribución y ventas de SAP. Las consultas habrían de poder ser variadas, asimismo las fallas en las funcionalidades o mejoras requeridas. Para minimizar la heterogeneidad de los incidentes entre los grupos de prueba se evaluó previamente cada

uno ya que la dificultad que plantean los incidentes puede variar así como la problemática que conllevan. A cada participante se le asignó 2 incidentes (6 en total por equipo).

Métricas: las variables dependientes en las que concentró el experimento fueron velocidad y calidad del trabajo. Para medir la velocidad se tomó el tiempo desde el momento en que fue asignado el incidente hasta que pudo ser resuelto. En lo que respecta a la calidad del trabajo se asignó puntos a cada equipo en función de los parámetros descritos en el cuadro 4.2.

Cuadro 4.2 Método de evaluación de la documentación

Concepto	Puntos
¿Hay notas de la evolución del caso en el incidente?	3
¿Se notifico al usuario de los avances?	2
¿Están los enlaces a los documentos de configuración y programación afectados?	2
Calidad de la documentación	1 a 10

Fuente: Elaboración propia

Se buscó medir la calidad del trabajo de cada equipo en cada incidente. Así, si en el incidente hay un log donde se ejemplifica que hay notas explicando los avances en la resolución se le otorgan 3 puntos, si adjunta pruebas (email por ejemplo) que se comunicó con el usuario se agregan 2 puntos, si señala los enlaces a los documentos de configuración se suman 2 puntos más y luego se mide la calidad de estos últimos. Para este aspecto se le pide al líder técnico del equipo que califique de 1 al 10 la documentación donde se detallan los cambios de configuración y programación en el sistema. Esta evaluación si bien puede ser considerada subjetiva es desarrollada por la persona que mas conocimiento tiene del grupo, el ser una sola persona se puede asegurar un único criterio para evaluar la documentación por lo que se evalúa de igual manera a todos.

Reglas especiales para el grupo *FTS*: antes de comenzar el experimento fueron capacitados en algunos aspectos de la metodología. Como ser la importancia del trabajo en equipo, el valor de documentar los avances y cambios, la criticidad de un buen *hand off* y el concepto de *CP* (ya que forman uno ellos). Cada uno debería entender cómo llevar a cabo un *hand*

off. Como ayuda se distribuyó una lista que deberían verificar cada vez que reciban uno, esto se ven en más detalle en el cuadro 4.3

Cuadro 4.3 Checklist durante los *hand off*

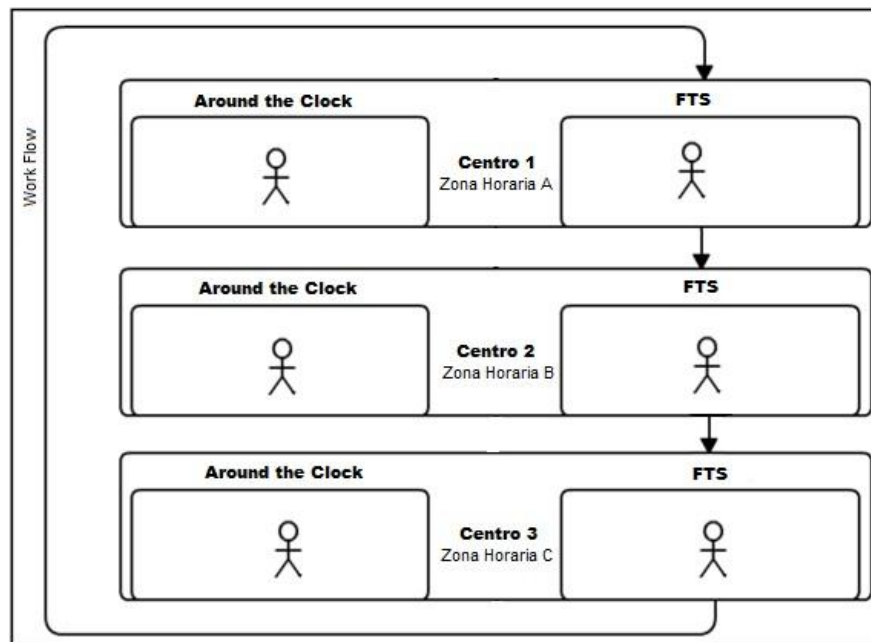
Acciones	Si / No	Notas
¿Fue la calidad de audio buena?		
¿Hablo en forma clara y pausada?		
¿Fue envió por email el <i>hand off</i> ?		
¿Se utilizó el formulario de <i>hand off</i> ?		
¿Fueron las tareas explicadas?		
¿Fueron contestadas las preguntas realizadas?		
¿Se compartió la pantalla?		
Duración de la reunión		

Fuente: Elaboración propia

También se puso énfasis en la importancia de realizar los *hand off* vía teléfono compartiendo la pantalla (via *Webex*), mientras mayor sea la interacción mejor. Además se llevaron a cabo trainings de diversidad explicando las culturas de cada uno de los países. Se utilizó este tiempo para que los miembros se conozcan un poco más y hablen entre ellos tanto de trabajo como experiencias personales.

Reglas para los grupos *ATC*. Estos siguieron trabajando de la misma manera que lo venían haciendo. No deberían realizar ningún solapamiento o recibir ningún tipo de comentario respecto a *FTS*. Se dedicarían solamente a resolver sus incidentes.

Figura 4.1 Distribución de los Equipos



Fuente: Elaboración propia

3. Resultados

Antes de comenzar a analizar los resultados es conveniente tener en consideración la envergadura y carácter meramente cualitativo y exploratorio del trabajo de campo realizado. Asimismo destacar la utilidad de ampliar la envergadura de esta corroboración empírica con otros trabajos que involucren una mayor cantidad de consultores, trabajando con el modelo durante un periodo más prolongado y aplicarlo con otros grupos de consultores funcionales asimismo buscando aislar otras posibles variables ocultas.

Las hipótesis planteadas en el estudio fueron tres:

H.1: Es posible aplicar metodologías *FTS*, hasta hoy utilizadas en desarrollo de software, a grupos de soporte de *ERP*

Esto pudo ser comprobado en el experimento. El mismo pudo ser llevado a cabo de manera exitosa.

H.2: Las metodologías *FTS* permitirían a los *ERP* (*Enterprise Resource Management*) resolver los numerosos problemas que presentan y que obedecen a la carencia de una metodología de gestión.

También esta pudo ser corroborada dentro del experimento. En función de los resultados expuestos en el cuadro 4.4 se ha podido identificar un alineamiento del grupo con la metodología *FTS*. Durante la semana se realizaron 15 *hand off*, en el cuadro se puede corroborar que los consultores indicaron que en la mayoría de las ocasiones siguiendo la metodología propuesta se lograron *hand off* claros y precisos que ayudaron a la correcta implementación de *FTS*.

Cuadro 4.4 Resultado de los *hand off*

Acciones	Resultados
¿Fue la calidad de audio buena?	14/15
¿Hablo en forma clara y pausada?	11/15
¿Fue envió por email el <i>hand off</i> ?	12/15
¿Se utilizó el formulario de <i>hand off</i> ?	15/15
¿Fueron las tareas explicadas?	14/15
¿Fueron contestadas las preguntas realizadas?	12/15
¿Se compartió la pantalla?	15/15
Duración de la reunión	Promedio 22 minutos

Fuente: elaboración propia

Del proceso de *hand off* se pudo observar grupos mejor formados en distintas temáticas. No solo trabajan en sus incidentes sino que a su vez ayudan a otros y se involucran intensamente en resolverlos. En lugar de solo especializarse en un tema atienden varios a la vez. Logran retomar el control del conocimiento del sistema, si bien al principio los participantes destacaron lo abrumador que puede ser al principio atender tantos temas la rigurosidad del proceso de *hand off* les otorgo una estructura que los guió en el día a día. Identifican como positivos los resultados finales.

Un comentario recibido en común, entre los distintos participantes del equipo *FTS*, fue que era vital para un buen funcionamiento contar con un nivel de ingles intermedio a avanzado, hablar pausado y tratando de reducir los acentos en lo posible. La barrera idiomática se

presentaba como un factor que no debía ser desconsiderado. Por último mencionaron lo importante que resultó el concepto de *CP* integrado al proceso. Las preguntas al final de cada turno (sección 2.3) ayudaron al equipo a identificar hacia donde debían dirigirse los esfuerzos.

H.3: La aplicación de metodologías *FTS* a los *ERP* (*Enterprise Resource Management*) permite resolver incidentes de forma más eficiente, en tiempos menores.

También esta hipótesis pudo ser corroborada dentro del experimento. Los resultados expuestos en el cuadro 4.5 y 4.6 indican que el equipo *FTS* no solo resolvió los incidentes de manera más rápida, sino que también su calidad de documentación fue mayor.

Cuadro 4.5 Tiempo de resolución de los incidentes

Sujeto	Equipo	Incidente	Tiempo de Resolución	Promedio Total por Equipo
S1	<i>FTS</i>	Incidente A	3 Días, 11 horas	3 Días, 17 horas
		Incidente B	3 Días, 14 horas	
S2		Incidente C	4 Días, 15 horas	
		Incidente D	3 Días, 17 horas	
S3		Incidente E	3 Días, 3 horas	
		Incidente F	3 Días, 21 horas	
S4	<i>ATC 1</i>	Incidente G	3 Días, 19 horas	4 Días, 8 horas
		Incidente H	5 Días	
S5		Incidente I	5 Días	
		Incidente J	3 Días	
S6		Incidente K	4 Días, 21 horas	
		Incidente L	4 Días, 12 horas	
S7	<i>ATC 3</i>	Incidente M	4 Días, 15 horas	4 Días, 11 horas
		Incidente N	3 Días, 18 horas	
S8		Incidente O	4 Días, 22 horas	
		Incidente P	4 Días, 19 horas	
S9		Incidente Q	5 Días	
		Incidente R	3 Días, 17 horas	

Fuente: elaboración propia

Cuadro 4.6 Calidad de la documentación

Sujeto	Equipo	Incidente	Puntaje	Promedio Total por Equipo
S1	FTS	Incidente A	17	16.33
		Incidente B	17	
S2		Incidente C	16	
		Incidente D	16	
S3		Incidente E	15	
		Incidente F	17	
S4	ATC 1	Incidente G	12	11
		Incidente H	11	
S5		Incidente I	10	
		Incidente J	11	
S6		Incidente K	12	
		Incidente L	10	
S7	ATC 3	Incidente M	13	11.16
		Incidente N	10	
S8		Incidente O	9	
		Incidente P	11	
S9		Incidente Q	12	
		Incidente R	12	

Fuente: elaboración propia

Es pertinente remarcar que los tiempos de reducción del equipo *FTS* respecto *ATC 1* fueron del 14% mientras que contra el equipo *ATC 2* fueron del 16%. Si bien los tiempos son menores están lejos de los valores teóricos imaginados en la introducción y en la sección 1.6.

Donde sí se ve un gran diferencial es en la calidad de la documentación. Las diferencias en las puntuaciones entre el equipo *FTS* y *ATC 1* fue del 48% y del 46% con *ATC 2*. Al parecer en este caso particular los efectos de *FTS* son más profundos en la calidad del trabajo y esparcimiento del conocimiento dentro de la organización que los tiempos de resolución de los incidentes.

CAPITULO 5. Conclusiones y propuestas

La baja adopción de estrategias *FTS* en el soporte *IT* no es una sorpresa, no hay prácticas y procesos que den soporte a las necesidades que una estrategia *FTS* conlleva. Para un mercado, que constantemente busca ventajas competitivas como reducción de costos y mayor eficiencia las operaciones *offshore* son cada vez más comunes. Este trabajo, puede facilitar la tarea en este sentido ya que el modelo presentado demostró, dentro del experimento, prometedoras ventajas del uso de *FTS* para el soporte de *ERP*.

Sin perjuicio de que resultaría muy aconsejable ampliar los alcances de la corroboración empírica de este estudio, todo parecería indicar que el *FTS* podría ser una buena solución para el soporte de los grupos de *ERP* y para que éste funcione de manera eficiente, una organización no solo debe prestar atención a los problemas técnicos. Debe focalizarse también en la manera en que los miembros de los diferentes equipos interactúan. El modelo propuesto muestra como:

- Las buenas prácticas y el cuerpo teórico de *FTS* ayudan a definir un proceso completo para aplicar *FTS*
- Como los subprocesos están conectados y como deben ser ejecutados

Los trabajos sobre *FTS* generalmente ponen su foco en la velocidad, muy pocos hablan de la calidad de las implementaciones o servicio provisto. Sin embargo, esta característica mostro dentro del experimento ser una de las principales virtudes de *FTS* para el modelo aplicado.

Otra variable que se debe considerar es la diversidad cultural entre los grupos. Los *Trainings* para integrar a los distintos grupos mostro ser una parte fundamental para alcanzar buenos resultados. Un fuerte lazo de comunicación y confianza entre las partes debe construirse si se desea llevar un proyecto de esta envergadura a cabo. Los equipos remotos deber ser motivados a siempre hacer preguntas y nunca realizar asunciones.

Se identificaron dentro del experimento distintos desafíos que deben ser tratados y que pueden afectar la performance de cualquier modelo *FTS*:

- Bajo nivel de inglés, puede volverse un obstáculo para compartir el conocimiento.
- Un criterio claro por el cual las locaciones sean seleccionadas. No se somete a una cuestión de costos, debe ser tal que se acople a la metodología *FTS*.
- Un proceso de *hand offs* durante las vacaciones y feriados
- Como lidiar con diferentes niveles de experiencia en cada centro, un nivel de experiencia desequilibrado puede afectar la productividad de un equipo. Puede causar problemas al momento del *hand off* ya que parten de distintos niveles de conocimiento, generando malentendidos.
- Falta de confianza entre los miembros de un equipo. Factores como la distancia o la cultura deben ser estudiadas en mayor profundidad.
- Aplicación de este modelo a distintas áreas de soporte IT

Hay ciertas limitaciones que este trabajo no cubre pero podrían ser el inicio de futuras investigaciones. Durante el experimento de campo la cantidad de personas que formaron parte del mismo ha sido baja. Esto es un impedimento para aplicar generalizaciones más abarcativas.

Otra oportunidad para seguir investigando sobre el tema es probar el modelo en distintas organizaciones, con diferentes idiosincrasias y valores, con otros tipos de organizaciones, otras tareas y con muestras más amplias analizar el comportamiento del modelo y su desempeño.

Las metodologías y métodos de trabajo son un campo excitante y en constante evolución. *FTS* es un claro ejemplo de esto, el aporte de este modelo está enfocado a soportar sistemas *ERP*. Pero dispara muchas preguntas que deberán ser contestadas en futuras investigaciones.

Bibliografía

Setamanit , Wakeland and Raffo. Exploring the impact of task allocation strategies for global software development using simulation. SPW/ProSim'06 Proceedings of the 2006 international conference on Software Process Simulation and Modeling. Pages 274-285 Springer-Verlag Berlin, Heidelberg ©2006

E. Carmel, J. A. Espinosa, and Y. Dubinsky, "Follow the Sun Workflow in Global Software Development", Journal of Management Information Systems Vol. 27 No. 1, pp. 17 – 38, 2010.

C. Visser and R. van Solingen. Selecting locations for follow-the-sun software development: Towards a routing model. IEEE International Conference on Global Software Engineering, pages 185–194, 2009.

E. Carmel and J. A. Espinosa, I'm Working While They're Sleeping: Time Zone Separation Challenges and Solutions, Kindle Edition, 2011, 188 p.

J. Kroll, S. I. Hashmi, I. Richardson, and J. L. N. Audy, "A Systematic Literature Review of Best Practices and Challenges in Follow-the-Sun Software Development", In Global Software Engineering Workshops (ICGSEW), pp. 18-23, 2013.

E. Carmel, Y. Dubinsky, and J. Johnston. Follow the sun workflow in global software development: Theory, modeling and quasi-experiment to explore its feasibility. In Proceedings of the Third Global Sourcing Workshop: The Impacts of Global IS Sourcing on Engineering, Technology and Innovation Management, Keystone, CO, USA, March 2009.

Gupta, A. Deriving mutual benefits from offshore outsourcing: The 24-Hour Knowledge Factory scenario, Communications of the ACM, 52, 6 (June, 2009), 122-126.

J. J. Treinen and S. L. Miller-Frost, "Following the Sun: Case Studies in Global Software Development," IBM Systems Journal, 45 (4), October 2006.

E. Carmel, Y. Dubinsky, and A. Espinosa. Follow the sun software development: New perspectives, conceptual foundation, and exploratory field study. Hawaii International Conference on System Sciences, 0:1–9, 2009.

C. K. Visser, "Connecting Time Zones and Languages in Follow-the-Sun: a routing model," Master's thesis, Delf University of Technology, Netherlands, 2009.

Gorton, I. and Motwani, S. Issues in Co-operative Software Engineering Using Globally Distributed Teams. Information and Software Technology, 38. 647-655. 2008

A. Gupta, E. Mattarelli, Seshasai S. and J. Broschak, "Use of collaborative technologies and knowledge sharing in co-located and distributed teams: Towards the 24-h knowledge factory," The Journal of Strategic Information Systems, Volume 18, 147-161, 2009.

[13] E. Carmel, "Building your Information Systems From the Other Side of the World: How Infosys manages time differences", MIS Quarterly Executive, 5(1), 2006.

Gupta, A, The 24-Hour Knowledge Factory: can it replace the graveyard shift? Computer, 42, 1 (January, 2009), 66-73.

J. Espinosa and E. Carmel. Model coordination costs due to time separation in global software teams. In Proceedings of the International Workshop on Global Software Development at the 25th International Conference on Software Engineering, pages 64–67, Portland, Oregon, 2003.

J. Espinosa and E. Carmel. The effect of time separation on coordination costs in global software teams: A dyad model. In HICSS '04: Proceedings of the 37th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, 2004.

K. Kumar, P. van Fenema, and M. von Glinow. Intense collaboration in globally distributed work teams: Evolving patterns of dependencies and coordination. *Managing Multinational Teams: Global Perspectives Advances in International Management*, 18:127–154, 2005.

J. Kroll, E. Hess, J. L. N. Audy, and R. Prikladnicki, “Researching into Follow-the-Sun Software Development: Challenges and Opportunities,” In: 6th International conference on Global Software Engineering (ICGSE), 2011, Helsinki, Finland.

E. Carmel. *Global Software Teams, Collaborating Across Borders and Time Zones*. Prentice Hall PTR, 1999.

M. Yap. Follow the sun: Distributed extreme programming development. In *ADC '05: Proceedings of the Agile Development Conference*, pages 218–224, Washington, DC, USA, 2005. IEEE Computer Society.

J.A. Espinosa, N. Nan, and E. Carmel “Do Gradations of Time Zone Separation Make a Difference in Performance? A First Laboratory Study”, *International Conference on Global Software Engineering*, Munich, Germany, August 27-30, 2007.

J. Kroll, R. Prikladnicki, J. L. N. Audy, E. Carmel, and J. Fernandez, “A Feasibility Study of Follow-the-sun Software Development for GSD Projects”, *International Conference on Software Engineering (SEKE)*, 2013, Boston, USA.

E. Hess and J. L. N. Audy, "FTSProc: A Process to Alleviate the Challenges of Projects that Use the Follow-the-Sun Strategy," *Global Software Engineering (ICGSE)*, 2012 IEEE Seventh International Conference on , vol., no., pp. 56-64, 27-30 Aug. 2012.

J. Kroll, A. R. Santos, R. Prikladnicki, E. R. Hess, R. Glanzner, A. Sales, J. L. N. Audy and P. Fernandes, “Follow-the-Sun Software Development: A Controlled Experiment to

Evaluate the Benefits of Adaptive and Prescriptive Approaches,” Proceedings of the 24th International Conference on Software Engineering & Knowledge (SEKE 2012), 551-556.

J. Kroll, I. Richardson, J. L. Audy, and J. Fernandez, J., “Handoffs Management in Follow the-Sun Software Projects: A Case Study”, In 47th Hawaii International Conference on System Sciences (HICSS), Hawai'i Island, USA, 2014.

V. R. Solingen and M. Valkema, “The Impact of Number of Sites in a Follow the Sun Setting on the Actual and Perceived Working Speed and Accuracy: A Controlled Experiment”, Global Software Engineering (ICGSE), 5th IEEE International Conference, 165- 174, 2010.

Kroll, J. and Audy, J. L. N., "Mapping Global Software Development Practices for Follow-the-Sun Process," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , pp. 164-168, 27-30 Aug. 2012.

G. Lee, W. DeLone, and J. A. Espinosa. Ambidextrous coping strategies in globally distributed software development projects. *Commun. ACM*, 49(10):35–40, 2006.

Espinosa, J. A., Cummings, J. N., Wilson, J. M., & Pearce, B. M., Team boundary issues across multiple global firms. *Journal of Management Information Systems*, 19, 4, (2003), 157-190.

Espinosa, J.A. and Carmel, E., Modeling the Effect of Time Separation on Coordination Costs in Global Software Teams. in *The 37th Hawaii International Conference on System Sciences*, (Hawaii, USA, 2003).

R. Czekster, P. Fernandes, R. Prikladnicki, A. Sales, A. Santos, and T. Webber, “Follow-The-Sun Methodology in a Stochastic Modeling Perspective,” in 6th IEEE International Conference on Global Software Engineering (ICGSE): Methods and Tools for

Project/Architecture/Risk Management in Globally Distributed Software Development Projects (PARIS), Helsinki, Finland, August 2011, pp. 54–59.

A. R. Santos, A. Sales and P. Fernandes, "Setting Up a Stochastic Model for Teams Working in a Follow-the-Sun Environment," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , vol., no., pp. 179, 27-30 Aug. 2012.

J. C. Tang, C. Zhao, X. Cao, and K. Inkpen, "Your time zone or mine?: a study of globally time zone-shifted collaboration", Proceeding of the ACM 2011 conference on Computer supported cooperative work (CSCW 2011). ACM, New York, NY, USA, 235-244, 2011.

Hazzan, O. and Dubinsky, Y., Agile Software Engineering, London: Springer-Verlag, 2008.

J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams". System Sciences, HICSS 2007. 40th Annual Hawaii International Conference on, p.274, 2007.

A. Gupta, S. Seshasai, and R. Aron, "Research Commentary: Toward the 24-Hour Knowledge Factory - A Prognosis of Practice and a Call for Concerted Research" (November 19, 2006). Eller College of Management Working Paper No. 1038-06 Available at SSRN: <http://ssrn.com/abstract=946012>

Kroll, J.; da Silva Estacio, B.J.; Audy, J.L.N.; Prikladnicki, R., "An Initial Framework for Researching Follow-the-Sun Software Development," Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on , vol., no., pp.116,124, 18-21 Aug. 2014

Kroll, J. and Audy, J. L. N., " Follow-the-Sun Strategy: A Process for Global Software Development" Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , pp. 76-78, 27-30 Aug. 2012.

Kroll, J. and Audy, J. L. N., "FTS-SPM: A Software Process Model for Follow the Sun Development: Preliminary Results" Global Software Engineering Workshop (ICGSEW), 2014 IEEE Seventh International Conference on , pp. 21-16, 18-18 Aug. 2014.

J. Kroll, I. Richardson, and J. L. N. Audy, "Proposing a Software Process Model for Follow the Sun Development", 26th International Conference on Software Engineering & Knowledge (SEKE), 2014, Vancouver, CA.

Espinosa, J.A. and Carmel, E. The Impact of Time Separation on Coordination in Global Software Teams: A Conceptual Foundation. Software Process Improvement and Practice, 8 (4). 249-266. 2003

Espinosa, J.A. and Carmel, E. Modeling coordination costs due to time separation in global software teams, International Workshop on Global Software Development, part of the International Conference on Software Engineering, Portland, Oregon, USA, IEEE, 2003, 64-68.

Espinosa, J.A., Slaughter, S.A., Kraut, R.E., and Herbsleb, J.D. Familiarity, complexity and team performance in geographically distributed software Development, Organization Science, 18, 4, (July-August, 2007), 613-630.

Saunders, C.S., Van Slyke, C. and Vogel, D. My time or yours? Managing time visions in global virtual teams, Academy of Management Executive, 18, 1, (2004), 19-31.

H. Zhang and B. Kitchenham, "Semi-quantitative Simulation Modeling of Software Engineering Process," in Proceedings of the International Software Process Workshop and International Workshop on Software Process Simulation and Modeling, vol. 3966. Shanghai, China: Springer-Verlag Berlin Heidelberg, 2006, pp. 242–253.

C. Visser. Follow-the-Sun Routing Model. <http://www.follow-the-sun.org>.

A. Cameron, "A Novel Approach to Distributed Concurrent Software Development using a "Follow-the-Sun" Technique.", Unpublished EDS working paper, 2004.

J. Herbsleb, J., et al. "An Empirical Study of Global Software Development: Distance and Speed", in 23rd. International Conference on Software Engineering (ICSE). 2001. Toronto, Canada: IEEE Computer Society Press.

E. Carmel and R. Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE Software*, 18(2):22–29, 2001.

E. Carmel and P. Tjia. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, June 2005.

S. McConnell. The 10 most powerful ideas in software engineering. Keynote Presentation at the International Conference on Software Engineering, 2009.

P. Sooraj and P. K. J. Mohapatra, "Modeling the 24-hour software development process," *Strategic Outsourcing: An International Journal*, vol. 1, no. 2, pp. 122–141, 2008.

J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "Distance, dependencies, and delay in a global collaboration," in *Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work (CSCW'00)*. Philadelphia, PA, USA: ACM, 2000, pp. 319–328.

N. Denny, I. Crk, R. Seshu, "Agile Software Processes for the 24-Hour Knowledge Factory Environment," *Journal of Information Technology Research*, Vol. 1, Issue 1, pgs 57-71, Jan 2008

H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, B. Fitzgerald, "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio- Cultural

Distance”. Proceedings of the IEEE international conference on Global Software Engineering (ICGSE '06). IEEE Computer Society, Washington, DC, USA, 2006, 3-11.

V. Casey, S. Deshpande and I. Richardson, “Outsourcing Software Development. The Remote Project Manager’s Perspective”. The 2nd Global Sourcing Workshop - Services, Knowledge and Innovation, Val D’Isere, France, 2008.

S. Deshpande and I. Richardson, “Management at the Outsourcing Destination - Global Software Development in India,” In Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering (ICGSE '09). IEEE Computer Society, Washington, DC, USA, 217-225, 2009.

N. Denny, S. Mani, R. Sheshu, M. Swaminathan, and J. Samdal, “Hybrid offshoring: Composite personae and evolving collaboration technologies,” Information Resources Management Journal, 21, 1, 2008, pp. 89–104.

SAP Sitio Oficial <http://searchsap.techtarget.com/definition/ERP>

Bidgoli, Hossein, (2004). The Internet Encyclopedia, Volume 1, John Wiley & Sons, Inc.

Rubina Adam, Paula Kotze, Alta van der Merwe. 2011. Acceptance of enterprise resource planning systems by small manufacturing Enterprises. In: In: Proceedings of the 11th International Conference on Enterprise Information Systems, Vol.1, edited by Runtong Zhang, José Cordeiro, Xuwei Li, Zhenji Zhang and Juliang Zhang, SciTePress.

Shaul, L. and Tauber, D. 2012. CSFs along ERP life-cycle in SMEs: a field study. Industrial Management & Data Systems.

Khosrow-Puor, Mehdi. (2006). Emerging Trends and Challenges in Information Technology Management. Idea Group, Inc.

Anderegg, Travis. 2013. "MRP/MRP II/ERP/ERM — Confusing Terms and Definitions for a Murkey Alphabet Soup".

Sheilds, Murell G. (2005). E-Business and ERP: Rapid Implementation and Project Planning. John Wiley and Sons, Inc.

Chang, SI; Guy Gable; Errol Smythe; Greg Timbrell (2000). A Delphi examination of public sector ERP implementation issues. International Conference on Information Systems. Atlanta: Association for Information Systems.

Shaul, L. and Tauber, D. 2013. Critical Success Factors in Enterprise Resource Planning Systems: Review of the Last Decade. ACM Computing Surveys.

Monk, Ellen and Wagner, Brett. "Concepts in Enterprise Resource Planning" 3rd.ed. Course Technology Cengage Learning. Boston, Massachusetts. 2009

Prasad Bingi; Maneesh K. Sharma and Jayanth K. Godla "Critical Issues Affecting an ERP Implementation". Information Systems Management 16 (3). Auerbach Publications. 1999.

Walsh, Katherine (January 2009). "The ERP Security Challenge". CSOnline. CXO Media Inc

Young, Joanna (2014). "Best-of-Breed vs. ERP: What's Best for Higher Ed Today?". www.evolllution.com. The Evolllution Inc.

A. N. Terlato, 2014, "Motivación y Productividad en Empresas Argentinas" Tesis Doctoral, UCEMA, Argentina.