

# PREVENCIÓN DE CONFLICTOS EN LA INTEGRACIÓN E IMPLEMENTACIÓN DE CONTRATOS INFORMÁTICOS

EDGARDO J. FITZPATRICK

COMPUTADOR CIENTÍFICO POR LA UNIVERSIDAD DE BUENOS AIRES Y DIPLOMADO EN DERECHO INFORMÁTICO POR LA UNIVERSIDAD DE CHILE. CONSULTOR INTERNACIONAL DE PROYECTOS INFORMÁTICOS.

SUMARIO: 1. Introducción. 2. La informática, el software y los servicios informáticos. Antecedentes. La calidad del software. Estadísticas de Standish Group. 3. Conceptos básicos de proyectos informáticos. Integración de sistemas. Contratos de integración de sistemas. Tipos de proyectos de integración de sistemas. Clara definición de requerimientos. Control de cambios. Énfasis en el proceso de pruebas. Evaluación y manejo de riesgos. 4. Problemas frecuentes en proyectos informáticos. Problemas contractuales. Problemas generales. Problemas técnicos. 5. Conclusiones. Acciones posibles. Conclusiones finales.

## RESUMEN

El autor, a partir de su experiencia profesional en proyectos informáticos internacionales de integración de sistemas, cataстра los puntos críticos de los mismos y las consecuencias que ellos acarrean, planteando soluciones dentro del ámbito metodológico de los proyectos informáticos en relación a los contratos informáticos y sus anexos técnicos.

## PALABRAS CLAVE

Contratos informáticos, proyectos informáticos, problemas comunes, servicios informáticos, flujos de información, anexos técnicos.

## ABSTRACT

The author, from its professional experience in international computer science projects of systems integration, registers the critical points and the consequences that they carry, raising solutions within the methodological scope of the computer science projects in relation to computer science contracts and their technical attachment.

## KEY WORDS

Computer science Contracts, computer science projects, common problems, computer science services, flows of information, technical attachment

## 1. INTRODUCCIÓN

La presente monografía se genera dentro del marco del III Diploma de Postítulo en Derecho Informático. La monografía se realiza en base a la experiencia del autor en Proyectos Informáticos Internacionales de Integración<sup>1</sup> de Sistemas en empresas de clase mundial.

Se pone énfasis en el déficit con el cual se brindan generalmente los servicios informáticos y la diversidad de problemas que se presentan y las consecuencias que ellos acarrean. Se pone foco en proyectos de desarrollo a medida y proyectos de Integración de Sistemas.

Finalmente se plantean soluciones dentro del ámbito metodológico de los proyectos informáticos como las consideraciones con las cuales se deben integrar los contratos informáticos y sus respectivos anexos técnicos.

También se incluye un breve detalle de situaciones para las que se deben negociar o peritar a fin de transmitir al lector las dificultades que se presentan en esta instancia y por ende la importancia de realizar un esfuerzo en la prevención de conflictos.

Las situaciones que se describen corresponden a proyectos informáticos de envergadura (promedio un millón de dólares). Justamente la magnitud de los mismos los convierte, en caso de conflictos, en situaciones usualmente a ser tratadas por abogados y peritos informáticos. La particularidad de los servicios informáticos es que están regidos en gran medida por el acuerdo de las partes formalizado en el Contrato de Servicios Informáticos. Por lo tanto las acciones de prevención de conflictos se deben concentrar en las actividades de preventa y cláusulas del contrato de servicio.

Se espera que esta monografía ayude, a los profesionales relacionados con la contratación de servicios informáticos de envergadura, a evitar conflictos y controversias a través de las acciones preventivas que se sugieren en el presente.

## 2. LA INFORMÁTICA, EL SOFTWARE Y LOS SERVICIOS INFORMÁTICOS

### ANTECEDENTES

La informática tiene pocos años de existencia y menos aún de haberse masificado; esto trae aparejado que sea una actividad que carezca de una normativa ampliamente conocida y que los estándares técnicos no sean utilizados de manera universal.

Durante los orígenes de este proceso, que podemos identificar hasta la década del 70, la informática se caracterizaba por la escasez. En este entorno, había pocos profesionales de informática, siendo la mayoría profesionales de otras ramas de la ingeniería. Estos profesionales emprendían proyectos que se asimilaban más a sueños, en los que pretendían satisfacer necesidades muy demandantes con recursos muy escasos y a veces poco confiables. Esto generaba procesos quijotescos en los cuales los equipos trabajaban casi sin dormir, durante semanas o meses, exigiendo su creatividad al máximo. Si lograban su propósito se consideraban héroes pero en general eran incomprendidos por el resto, ya sea porque los demás no lograban comprender las adversidades que habían tenido que superar, o porque ellos sentían su logro liberado de otros factores como plazo de entrega, presupuesto o requerimientos del usuario, factores con los que obviamente el entorno los juzgaba.

<sup>1</sup> Ver sección describiendo Integración de Sistemas.

El éxito de los proyectos y de las organizaciones informáticas tenía una dependencia total en la capacidad de sus equipos humanos y en muchas ocasiones los éxitos se conseguían de la mano de líderes tecnológicos que muchos consideraban genios locos. En este contexto se solía mencionar con cierto orgullo que programar era un arte<sup>2</sup> (no confundir con la obra *The art of Computer Programming*, obra emblemática de Donald KNUTH que con fuerte contenido matemático da sustento a la programación), como todo arte se trataba de un proceso impreciso que generaba un producto con errores.

Hasta la década del 70, la informática se relacionaba directamente con las máquinas y su capacidad y el software, principalmente el sistema operativo y demás programas accesorios, eran un complemento que proveía la empresa que vendía el hardware. Pero lentamente, a partir de los sistemas que permitieron multiprocesamiento (como el Sistema 360 de IBM) y la disponibilidad de más capacidad de proceso, el Software comenzó a evolucionar como nunca antes, a adquirir otra dimensión y a cobrarse por separado. En un momento IBM llegó a pronosticar que próximamente la mitad de sus ingresos provendrían del Software. Sobre fines de la década del 80 sucedió algo similar con los servicios, hasta ese momento los servicios informáticos provistos por los grandes fabricantes de Hardware eran un complemento necesario para vender máquinas. Pronto IBM volvió a pronosticar que en pocos años la mitad de sus ingresos provendrían de los servicios.

Poco a poco la dependencia de las grandes industrias, empresas y Estados sobre la informática comenzó a crecer. Había cierta frustración pues la calidad de las soluciones informáticas era baja. Mientras que el equipamiento (hardware) se desarrollaba y producía utilizando estándares de ingeniería y procesos mejorados de acuerdo a las filosofías de calidad de la época, el Software parecía imposible de transformarse en una rama seria de la ingeniería.

Hubo una serie de eventos clave que permitieron mejorar el proceso de desarrollo de software, citando a continuación los más significativos:

- 1) ACM (Association for Computing Machinery) fundada en 1947, entre muchos esfuerzos instituyó las conferencias del ICSE (International Conference on Software Engineering).
- 2) Desarrollos de métodos formales para el desarrollo de Software Edsger Dijkstra.
- 3) Investigaciones en análisis estructurado y diseño de software por Tom DeMarco, Larry Constantine y Edgard Yourdon.
- 4) Contribuciones de Fred Brooks incluyendo "The Mythical Man-Month" a finales de los 80 y "No Silver Bullet".
- 5) Una serie de contribuciones pioneras de Peter Freeman y Tony Wasserman (1976) Industry/Academia "Interface Conference" y los primeros trabajos sobre educación en el ICSE (1976 y 1978).
- 6) Los esfuerzos conjuntos del IEEE-CS y del ACM fundando el Software Engineering Coordinating Comité, su Proyecto de Educación (SWEEP), su grupo de trabajo (WGSEET) y su libro de conocimiento (SWEBOK).
- 7) Contribuciones a la programación estructurada y el algoritmo de desarrollo por Knuth, Dijkstra, Wirth, Hoare y otros (entre 1966 y 1971).
- 8) Conferencias de la OTAN en Ingeniería de Software (1968-69).

<sup>2</sup> *The art of Computer Programming*, obra de 4 volúmenes publicados progresivamente a través de los años. Nos referimos aquí al volumen 1, tercera edición, de junio de 1997, de la editorial norteamericana Addison-Wesley. Véase también sitio web <http://www-cs-faculty.stanford.edu/~knuth/taocp.html> [Consulta: 20.04.2006]

- 9) Los primeros programas de Master en Ingeniería de Software del Instituto Wang, la Universidad de Seattle y TCU (1978-79).
- 10) Desarrollo corporativos en gran escala de programas de capacitación entre los que se destacan los de IBM y Motorola University (finales de los años 70).
- 11) Solicitud del Departamento de Defensa de EEUU a sus proveedores de una metodología para desarrollo de Software<sup>3</sup>.
- 12) PMI (Project Management Institute) que desde 1984 certifica profesionales en desarrollo de proyectos.
- 13) Las contribuciones de la Universidad de Carnegie Mellon y su instituto de Ingeniería de Software (SEI) que crearon el CMM<sup>4</sup> (1991).

Hemos comentado entonces que al comienzo los proyectos informáticos giraban mayoritariamente alrededor del Hardware, luego el Software pasó a tomar mayor preponderancia. Finalmente los servicios informáticos fueron ganando terreno y desplazando al Hardware y al Software.

En forma paralela el conocimiento de informática fue cambiando: al principio solo había grandes computadores y pocas personas involucradas en el mundo informático. Luego a principios de 1980 nace el computador personal; Apple, IBM, Microsoft e Intel son los líderes que masifican la tecnología, por último a fines de los 90 Internet masifica el uso de computadores ofreciendo compartir el conocimiento en forma universal.

#### LA CALIDAD DEL SOFTWARE

Como se mencionó anteriormente, el software era asociado con baja calidad. Peter FREEMAN<sup>[5]</sup> denominó esta situación la crisis del software de las últimas dos décadas (70 y 80) por ser el responsable de presupuestos y plazos excedidos, baja calidad y expectativas no razonables. En el prólogo referenciado anticipaba que a pesar que muchas de las consecuencias continuaban estando presentes, había un punto de inflexión pues en dicho momento ya se consideraba que bajo las condiciones apropiadas, el proceso de software era tan manejable como tantas otras cuestiones que el hombre había aprendido a manejar en su universo.

Peter FREEMAN se estaba refiriendo en general al proceso de ingeniería de software cuyos principales contribuidores ya han sido mencionados y en especial al proceso de desarrollo de software.

El proceso de Ingeniería de Software converge durante la década del 80 y 90 con las industrias y empresas que emprendieron procesos de calidad acompañados por una serie de teorías, premios y estándares de certificación. Entre ellos podemos citar:

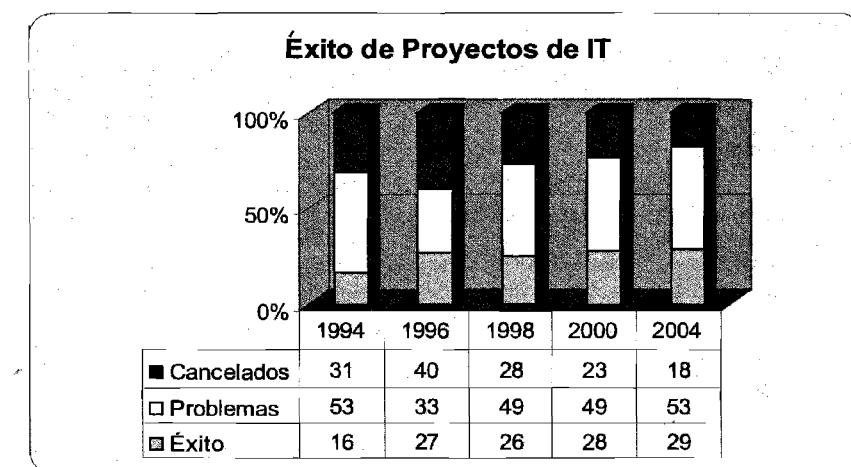
- o TQM de Deming, que a pesar de tener origen en 1940, tuvo furor en EEUU a partir de 1984.
- o Premio de EEUU Malcom Baldrige Nacional Quality Award a partir de 1987.
- o Proceso Six Sigma nacido en Motorola a mediados de los 80.
- o El estándar de calidad Inglés (British Standard 5750) a partir de 1979.
- o El estándar europeo ISO 9000 a partir de 1987.

Con todos estos antecedentes uno se imaginaría que los problemas del Software y de los proyectos informáticos que desarrollaban y/o integraban software se habrían desvanecido para el final del milenio, pero como se verá en los próximos párrafos, esto no fue así.

#### ESTADÍSTICAS DE STANDISH GROUP

La evolución de la calidad de los proyectos de software pueden consultarse en los informes Chaos del Standish Group<sup>6</sup>. Este grupo formado en 1985 en Massachusetts, ha recopilado información sobre el éxito y fracaso de proyectos de software. Hoy en día lleva acumulada información sobre 50.000 proyectos de IT, 58% originarios de EEUU, 27% corresponden a Europa y el 15% restante al resto del mundo. Representan en un 45% a empresas grandes del Fortune 1000, 35% de las empresas son medianas y 20% a empresas chicas.

En el siguiente gráfico<sup>7</sup> se pueden ver la cantidad de proyectos con éxito, con dificultades (tardaron mucho más y costaron mucho más) y los proyectos cancelados.



<sup>3</sup> El autor participó de un seminario dictado por James TOMAYKO en Buenos Aires, en 1992, en donde se entregó como antecedente el requerimiento del Departamento de Defensa de Estados Unidos de solicitarle a sus proveedores sus propuestas para contar con una metodología de desarrollo del Software a fin de seleccionar una que pasaría a ser un estándar. Aparentemente la metodología elegida fue la de Boeing.

<sup>4</sup> CMM (Capability Maturity Model) método que permite evaluar el nivel de madurez del proceso de desarrollo de Software de una Organización. Modelo que surgió a partir de una solicitud del Departamento de Defensa de EEUU para poder evaluar a sus proveedores.

<sup>5</sup> Prólogo de *Managing the Software Process*, de Watts HUMPHREY, libro editado en EE.UU. por Addison-Wesley Publishing Company Inc. en agosto de 1990 bajo el ISBN 0-201-18095-2.

<sup>6</sup> El resumen de los informes hasta el año 2000 se puede consultar en la siguiente dirección: [http://www.standishgroup.com/sample\\_research/PDFpages/extreme\\_chaos.pdf](http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf) [Consulta: 20.04.2006]

<sup>7</sup> Los datos correspondientes al 2004 se encuentran en [http://www.standishgroup.com/sample\\_research/PDFpages/q3-spotlight.pdf](http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf) [Consulta: 20.04.2006]

El Éxito de los Proyectos de IT se basa en cumplir con tres metas fundamentales:

- Alcance: El cliente termina recibiendo toda la funcionalidad que esperaba y con el nivel de calidad esperado
- Costo: El cliente termina pagando los montos presupuestados
- Tiempo: El cliente recibe la funcionalidad en el plazo acordado.

Pero antes de referirnos a los diversos problemas de los proyectos Informáticos y cómo estos criterios de éxito se ven impactados, será necesario explicar algunos conceptos relacionados a los proyectos informáticos a los cuales nos referiremos.

### 3. CONCEPTOS BÁSICOS DE PROYECTOS INFORMÁTICOS

#### INTEGRACIÓN DE SISTEMAS

Un proyecto informático de Integración de sistemas consiste en combinar productos y servicios de información en un Proyecto que provea una solución informática que resuelva un problema específico del cliente, resultando un solo Proveedor responsable del proyecto, rol conocido como Integrador de Sistemas.

Los componentes de la solución pueden ser:

- Hardware (de la marca del proveedor y/o de otras marcas)
- Software de Base (de la marca del proveedor y/o de otras marcas)
- Software Aplicativo (de la marca del proveedor y/o de otras marcas)
- Servicios (Desarrollo de Software Aplicativo, Soporte, Cableado, Telecomunicaciones, etc.)
- Otros productos y/o servicios

Como Integrador de Sistemas, el proveedor asume el rol de contratista primario y asume la responsabilidad de gestionar el proyecto desde su inicio hasta su finalización satisfactoria. El Proveedor es el único responsable ante el cliente por los servicios y productos referentes al proyecto, tanto del proveedor como de sus subcontratistas, simplificando y mejorando las vías de comunicación. La coordinación y la responsabilidad del resultado del proyecto está en manos del Gerente de Proyecto del proveedor.

Normalmente una Integración de Sistemas contiene, entre los productos y servicios a proveer, el desarrollo y/o adaptación de algún Software Aplicativo.

Los proyectos de este tipo requieren una gran involucración y participación por parte del cliente. Por la naturaleza del proceso de desarrollo y/o adaptación de Software Aplicativo, rara vez los requerimientos que dichos aplicativos deben satisfacer están totalmente claros al inicio del proyecto, siendo necesaria la definición de los mismos por parte del cliente. Más adelante se brindará mas detalle sobre este particular.

Para cada uno de los productos y/o servicios a proveer, se definen al inicio del proyecto Criterios de Aceptación, que establecen clara y objetivamente tanto para el proveedor como para el Cliente, las condiciones que deben cumplirse para que un producto y/o servicio sea aceptado.

Como cualquier proyecto, un proyecto de Integración de Sistemas puede verse afectado por factores externos al mismo (cambios en las necesidades del cliente por razones de merca-

do, cambios regulatorios o legales, cambios en la tecnología disponible). Para estas situaciones existe un proceso de Control de Cambios, de manera de poder registrar y manejar los cambios o alteraciones a lo definido con anterioridad. Más adelante se brindarán detalles sobre este proceso en particular.

#### CONTRATOS DE INTEGRACIÓN DE SISTEMAS

Debido a la variedad de productos y servicios combinados que se proveen, los contratos para este tipo de proyectos informáticos son un conjunto de documentos: Contrato Principal, Suplementos para servicios individuales, Anexos Técnicos que detallan aspectos de la solución contratada. La estructura de estos contratos varía de proveedor en proveedor por lo que habrá que revisar todos los documentos en forma minuciosa, identificando si todos los productos, servicios y condiciones están detallados y comprobar que estén claramente definidos los alcances de cada cláusula en el sentido de si aplican a todo el contrato o un producto o servicio en particular.

Es frecuente que las ofertas, propuestas y/o cotizaciones de Proyectos se realicen en un formato de presentación y sobre dicho documento se llegue a acuerdos entre el proveedor y cliente aceptando el proyecto. Dicho documento tiene peso legal y contractual pero no exime al cliente de la responsabilidad de confirmar en el contrato definitivo que se suscribe, que se respeta el alcance y términos acordados en la propuesta.

#### TIPOS DE PROYECTOS DE INTEGRACIÓN DE SISTEMAS

Existen diferentes tipos de Proyectos de Integración de Sistemas de acuerdo a las responsabilidades que el proveedor asuma en el mismo. La clasificación es la siguiente:

- Tiempo y Materiales: no existe por parte del proveedor compromiso más allá de proveer los productos y/o servicios contratados. Para el caso de provisión de Servicios, los mismos son aceptados por el cliente con el solo paso del tiempo. Si bien existe un objetivo de plazo para la ejecución del proyecto y un precio estimado, ambos, tiempo y precio permanecen indeterminados hasta que el proyecto es completado.
- Plazo Fijo: similar al anterior, pero el proveedor se compromete a proveer productos y/o servicios en plazos predeterminados por contrato. Por lo tanto, el tiempo total de ejecución del proyecto se conoce al inicio del mismo.
- Plazo Fijo y Precio Fijo: en este caso el proveedor se compromete a proveer el conjunto de productos y/o servicios en plazos predeterminados por contrato y a un precio también predeterminado por contrato. Así, se conoce al inicio del proyecto el tiempo total de ejecución y el precio total del mismo.

#### CLARA DEFINICIÓN DE REQUERIMIENTOS

Un Proyecto de Integración de Sistemas puede verse, como la fabricación de un producto (una solución) para ser implementada y así resolver un cierto problema del cliente. Como en todo proceso de fabricación, la clara definición de las especificaciones del "producto a fabricar" es una tarea imprescindible y fundamental.

Como se mencionó anteriormente, un proyecto de Integración de Sistemas incluye el desarrollo y/o adaptación de algún Software Aplicativo. Los requerimientos (qué es lo que el Software Aplicativo debe hacer) deben ser brindados por los usuarios del cliente en la etapa de Definición/

**Validación de Requerimientos.** Será una Definición de Requerimientos en aquellos casos en que no se conozcan antes del inicio del proyecto los mismos. Será una Validación de Requerimientos en aquellos casos en que se tenga un entendimiento de los mismos antes de iniciar el proyecto, y los requerimientos deban ser verificados como válidos como primera etapa del proyecto.

En cualquier caso, los requerimientos deben ser relevados y documentados apropiadamente en un Documento de Requerimientos. Este documento pasa a ser uno de los más importantes del proyecto, debiendo controlarse y documentarse cualquier modificación a los requerimientos en el mencionado, a través del proceso de Control de Cambios.

Debe destacarse que este proceso, al ser el que determina las especificaciones que tendrá el producto final, es altamente complejo y crítico, por lo cual deben asignarse para su realización los mejores recursos, tanto del Proveedor como del Cliente.

## CONTROL DE CAMBIOS

Los proyectos Informáticos de envergadura generalmente tienen una duración prolongada y a veces superan varios años. Durante un período de esa extensión, es altamente probable que factores externos al mismo (como por ejemplo regulaciones, leyes, usos y costumbres) cambien y por lo tanto afecten de una u otra manera los productos o soluciones que el proyecto debe producir.

Por otro lado, cada proyecto es único, ya que uno de sus mayores insumos: la gente, es siempre diferente de un proyecto a otro. Esto aumenta las posibilidades de errores humanos, fallas de comunicación o interpretación que, una vez detectados, provocan alteraciones en definiciones, especificaciones o acuerdos realizados con anterioridad.

Por último, gran cantidad de proyectos informáticos no han alcanzado sus objetivos o simplemente han sido cancelados debido a los frecuentes cambios que, al no ser manejados y controlados adecuadamente, provocaron sobrecostos y/o demoras inaceptables.

El Control de Cambios es, junto con una clara definición de requerimientos, el concepto de mayor importancia en un proyecto Informático de desarrollo de Software.

Como se mencionó, un proyecto Informático de envergadura puede compararse a un proceso de fabricación, donde el producto a ser fabricado es una solución que resolverá un problema del cliente. Como todo producto a ser fabricado, esta solución tiene *especificaciones* o *requerimientos* que debe cumplir. Estas especificaciones son definidas en etapas muy tempranas del proyecto. Cualquier modificación a estas especificaciones alterará en mayor o menor medida el producto o solución a ser fabricada.

Estas modificaciones deben ser manejadas y controladas a través de la disciplina de Control de Cambios. Esta disciplina involucra una serie de procedimientos, formularios, revisiones y autorizaciones con el objetivo de conocer y mantener controlados tanto por el Proveedor como por el cliente las modificaciones a las especificaciones que existen con frecuencia en cualquier proyecto de este tipo.

Para una mayor comprensión del tema, es necesario aclarar cuándo se produce un cambio.

Las especificaciones o requerimientos de la solución a fabricar se registran en un documento denominado *Documento de Definición de Requerimientos*. En este documento se menciona, conceptualmente, qué es lo que el sistema debe hacer, y es uno de los "Documentos Base" del proyecto. En el proceso de definición de requerimientos interviene, como se mencionó anteriormente, el cliente (a través de los diferentes funcionarios de sus departamentos o usuarios), quienes especifican los requerimientos y personal del proveedor quienes deben asegurarse de comprender

los requerimientos y que los mismos queden adecuadamente documentados. El Documento de Requerimientos es entonces revisado y aprobado tanto por el cliente como por el proveedor. Cualquier modificación a las especificaciones documentadas (supresión, agregado o modificación) debe ser documentado, analizado, evaluado su impacto (un cambio puede alterar plazos y/o costos y/o recursos) y finalmente aprobado o rechazado por el nivel gerencial adecuado.

Una vez completada la etapa de Definición de Requerimientos, se ingresa en la etapa de Diseño Externo/Interno donde se determina cómo el sistema debe cumplir con los requerimientos o especificaciones documentadas y aprobadas en la etapa anterior. El producto de dicha fase es el Documento de Diseño Externo / Interno, que es otro de los "Documentos Base". Como todo "Documento Base" cualquier modificación a lo en él registrado debe ser manejada a través de la disciplina de Control de Cambios.

## ÉNFASIS EN EL PROCESO DE PRUEBAS

Los productos a ser fabricados en los proyectos Informáticos tienen, como se mencionó anteriormente, especificaciones que establecen qué es lo que el producto debe hacer, cómo debe hacerlo y que tan bien debe hacerlo.

El objetivo del proyecto es fabricar estos productos en el tiempo acordado, dentro de los presupuestos y alcanzando los niveles de calidad comprometidos. Estos niveles de calidad deben demostrarse.

La calidad en estos proyectos tiene dos dimensiones:

- Calidad del Proceso (standards, metodologías, procedimientos y controles).
- Calidad del Producto (el producto alcanza sus especificaciones).

La calidad se demuestra mediante:

- Pruebas Estáticas (revisiones e inspecciones de procesos y productos).
- Pruebas Dinámicas (verificación, mediante el ingreso de datos al sistema, del comportamiento del mismo versus un conjunto de resultados esperados).

El proceso de pruebas se inicia junto con el proyecto y usualmente tiene asignado un Gerente de Pruebas. Este proceso va en paralelo con el desarrollo / adaptación de aplicaciones y tiene sus tareas integradas en el plan del proyecto, junto con sus propios recursos, procedimientos y estándares. Este grupo está compuesto por perfiles de distintos tipos, significativamente diferentes a los perfiles que se encuentran en un grupo de desarrollo de aplicaciones. Los perfiles típicos son, entre otros: Representantes de Usuarios Finales, personal de Organización y Métodos, Auditoría, etc.

A diferencia de otras metodologías, aquí el proceso de pruebas comienza junto con la definición de requerimientos. En esta etapa, el Grupo de Pruebas comienza a analizar los requerimientos que son relevados. Para este grupo, los requerimientos son la base de su estrategia de pruebas y deben asegurarse que todo requerimiento relevado y registrado podrá ser probado / demostrado al cliente en las diferentes etapas de prueba. Puede llegarse a la situación en que un requerimiento es eliminado o no aceptado debido a que no puede establecerse una estrategia de prueba para el mismo.

El objetivo de las diferentes etapas de prueba es encontrar errores. Los errores se van introduciendo a lo largo de todo el proceso de fabricación de aplicaciones, ya que en su gran mayoría pueden considerarse como procesos *manuales de traducción* de una representación

del sistema (por ejemplo, el diseño interno) a otra representación del sistema (por ejemplo, los programas).

Glenford Myers<sup>8</sup> define:

- caso de prueba exitoso: es aquel que descubre un error
- caso de prueba no exitoso: aquel que no descubre un error

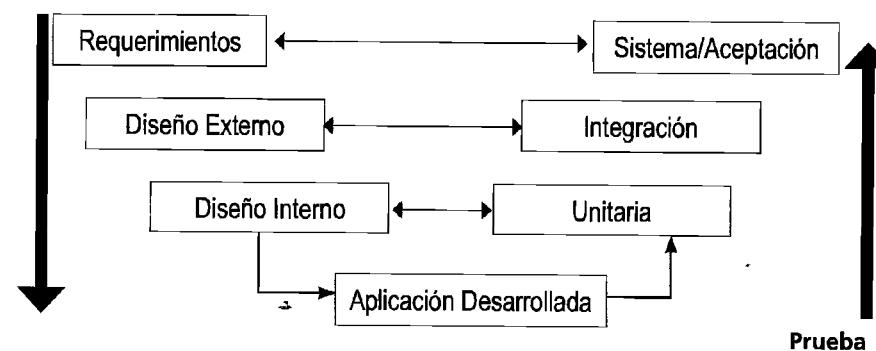
Por otro lado, los errores deben ser encontrados rápidamente, lo más tempranamente posible luego de haber sido introducidos. Esto se debe a que el esfuerzo necesario para corregir un error va creciendo rápidamente a medida que avanza el proceso de desarrollo. Como ejemplo, un error introducido en la etapa de relevamiento que podría haber sido corregido en esa etapa a un costo de 1 hora/hombre, tiene un costo de corrección de más de 300 horas/hombre si es detectado y corregido una vez que el sistema fue implementado.

Existen diferentes etapas o niveles de pruebas dinámicas a lo largo del proyecto. Típicamente, éstas son:

- Prueba Unitaria. Es aquella que se realiza sobre programas o módulos. Su objetivo es encontrar situaciones en donde los programas no se comportan de acuerdo a su Diseño Interno.
- Prueba de Integración. Se realiza sobre conjuntos de programas, subsistemas, o conjuntos de sistemas. Su objetivo es encontrar situaciones en donde los mencionados componentes no se comportan de acuerdo con su Diseño Externo.
- Prueba de Sistema. Es realizada sobre el sistema completo en un ambiente controlado. Se buscan situaciones en donde el sistema no se comporta de acuerdo a sus requerimientos.

El siguiente gráfico muestra la relación entre las diferentes etapas de fabricación y de prueba:

#### Fabricación



Para cada una de estas etapas se requiere:

- La definición y acuerdo con el cliente de los criterios de entrada y aceptación.

<sup>8</sup> Glenford Myers, *El Arte de Probar el Software*. Editorial el Ateneo, Buenos Aires, 1983.

- La elaboración de un plan de trabajo detallado, con los lineamientos indicados anteriormente.
- La elaboración de un lote de prueba, que debe incluir el resultado esperado para cada caso de prueba.
- Para cada caso de prueba, debe definirse si el mismo será verificado mediante:
- Análisis (examen matemático de algoritmos o modelos)
- Inspección (examen visual de componentes o documentación)
- Demostración (observación del comportamiento del sistema)
- Prueba (ejecución de un caso de prueba)
- La ejecución de todos los casos de prueba.
- El análisis y verificación de los resultados brindados por el sistema.
- El registro de todos los problemas encontrados, de acuerdo a un proceso documentado y acordado con el cliente y que permita controlar fácilmente el grado de avance en el proceso de pruebas.
- La corrección, salvo acuerdo con el cliente, de todos los problemas encontrados.
- Un estricto control y diferenciación de errores y cambios, ya que frecuentemente se reportan como errores situaciones que son en realidad cambios a las especificaciones.

Como conclusión, el énfasis que pone la metodología en el proceso de pruebas contribuye, en gran medida, a asegurar que los productos comprometidos serán entregados al cliente y aceptados por éste en los plazos previstos y con la calidad comprometida.

#### EVALUACIÓN Y MANEJO DE RIESGOS

Durante la preparación de un proyecto de Integración de Sistemas se realiza una Evaluación de Riesgos del Proyecto. La evaluación se realiza con la ayuda de una Guía de Evaluación de Riesgos, que lista los hechos o condiciones riesgosas que pueden presentarse en un proyecto de integración de Sistemas. Por ejemplo una lista puede constar de más de 20 puntos, con un promedio de 4 ítems por cada punto, abarca situaciones tan variadas como:

- Desarrollo de *hardware ad-hoc* para el proyecto
- Nivel de definición de requerimientos
- Experiencia del grupo en proyectos similares
- Experiencia del gerente de proyecto en proyectos similares
- Conexiones con sistemas de distintos proveedores
- Requerimientos de *performance*
- Nivel esperado de cambios
- Rigidez y/o criticidad del cronograma
- Involucración y dependencias del cliente en el proyecto
- Involucración y dependencias de los subcontratistas en el proyecto

- Términos y condiciones del contrato
- Criterios de aceptación
- Nivel de planeamiento del proyecto

Una vez identificados los riesgos, para aquellas situaciones que implican un riesgo medio o alto, se desarrolla un *Plan de Contención*. En este plan se detallan las acciones a tomar (que pueden implicar incluir en el presupuesto del proyecto costos adicionales, incluir en el contrato cláusulas especiales, redefinir criterios de entrada o aceptación, etc.) para mantener el riesgo en un nivel bajo o, si es posible, eliminarlo.

Durante la ejecución del proyecto, periódicamente se revisa la Evaluación de Riesgos, ya que por el propio desarrollo del proyecto pueden aparecer situaciones que impliquen riesgos no detectados oportunamente o, por el contrario, algunos riesgos pueden haber desaparecido.

Todas estas evaluaciones y controles de riesgos periódicas se deben revisar y documentar apropiadamente.

#### 4. PROBLEMAS FRECUENTES EN PROYECTOS INFORMÁTICOS

En base a la experiencia profesional del autor se detallan unos 100 problemas que acontecen en proyectos informáticos de envergadura. Se han tenido en cuenta únicamente proyectos de Desarrollo de Software a Medida y proyectos de Integración de Sistemas.

Se han agrupado en tres grupos, los problemas relacionados con los contratos (20), los problemas originados en cuestiones técnicas (40) y otros problemas generales (40). Primero se listarán los problemas y posteriormente se describirá brevemente cada uno.

##### PROBLEMAS CONTRACTUALES

Los siguientes problemas se han denominado "Contractuales" porque para evitarlos se debe incluir texto específico dentro de las cláusulas del contrato o como mínimo identificar en dichas cláusulas que el detalle se encuentra en un anexo al contrato:

1. No tener definido quién es la contraparte en el cliente
2. No tener definido qué se entregará
3. No tener definido cómo se acepta el proyecto
4. No tener definido cuándo se entregará algo
5. No tener definidos niveles de soporte
6. No tener definida la capacitación
7. No cuantificar todo
8. No requerir y disponer de Infraestructura de trabajo: espacio físico, sillas, escritorios, PC, teléfonos
9. No contar con acceso fuera de horario normal de oficina
10. No especificar el costo de horas adicionales
11. No contar con la propiedad del código o de las fuentes

12. No especificar soporte postproducción
13. No especificar mantenimiento
14. Realizar cotizaciones incompletas
15. No especificar cómo tratar los temas relativos a la confidencialidad
16. No considerar aspectos impositivos y tributarios del proyecto
17. No determinar qué hacer ante problemas con personas que participan
18. Composición del equipo de trabajo
19. Rotación del equipo
20. No prevenir los fracasos

#### DESCRIPCIÓN DE LOS PROBLEMAS CONTRACTUALES

No tener definido quién es la contraparte en el cliente

DESCRIPCIÓN	
Es común emprender un proyecto sin que se determine en el contrato de servicios que es necesaria la participación activa de personal del cliente y en especial que deberá existir una persona en la organización del cliente que tenga la responsabilidad del proyecto. Es fundamental que dicha persona disponga de tiempo para participar en una cantidad de actividades relacionadas con el proyecto, entre las cuales podemos citar:	
<ul style="list-style-type: none"> <li>○ Reuniones de avance de proyecto</li> <li>○ Revisiones y aprobación de cambios y definiciones que genera el desarrollo del proyecto</li> <li>○ Resolución de conflictos que se generan en la organización del cliente y afecten al proyecto</li> </ul>	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo y Costo: Nadie toma responsabilidad por resolver problemas originados en la organización cliente</li> <li>○ Alcance: Se desarrolla el proyecto sin contar con aprobación a los cambios y definiciones, por lo que es difícil que exista acuerdo y satisfacción por el producto final entregado</li> </ul>	
PREVENCIÓN	
Especificar claramente en el Contrato de Servicios la necesidad de contar con la participación de personal del cliente en el proyecto, indicar el tiempo de participación requerida, los roles y responsabilidades que tendrán.	
Ejemplo	
<p><i>Responsabilidades del cliente</i></p> <ul style="list-style-type: none"> <li>○ Designar a un responsable de su organización que</li> <li>○ Deberá aprobar el diseño, los cambios de alcance al proyecto, ...</li> <li>○ Será responsable de resolver los problemas y conflictos de la organización que ...</li> <li>○ Participará en las reuniones de comité de Proyecto ....</li> <li>○ Decidirá en representación de la Organización sobre las alternativas .....</li> </ul>	

- Designar a Representantes Usuarios a tiempo completo en las siguientes actividades del proyecto:
  - Participarán en las reuniones de relevamiento de requerimientos, informando ....
  - Revisarán los productos de diseño, prototipos, resultados de prueba de sistemas
  - Realizarán la prueba de aceptación de usuario
- Designar a personal de sistemas con dedicación a medio tiempo para
  - Adaptar las interfaces de los sistemas existentes
  - Garantizar el correcto funcionamiento del equipamiento utilizado para el proyecto
  - Realizar las tareas de instalación que por motivos de seguridad estén restringidas

#### No tener definido qué se entrega

DESCRIPCIÓN
Lo Obvio es el enemigo número 1 de los contratos informáticos. Se pone mucha atención en todo tipo de cláusulas y detalles, pero al momento de describir el servicio a proveer y/o el sistema que se entregará, se omiten muchos detalles que se consideran sobrentendidos pero al momento de entregar lo comprometido surgen las diferencias.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Alcance: Casi nunca se logrará satisfacer las expectativas del cliente si los servicios y sistemas a entregar no fueron acordados en forma detallada.</li> </ul>
PREVENCIÓN
Especificar claramente en el Contrato de Servicios cada uno de los servicios, en qué consisten, qué duración tendrán en el tiempo, qué elementos se entregarán y cuándo.
Ejemplo:
<p><b>Entregables</b></p> <ul style="list-style-type: none"> <li>○ Sistema de Facturación, funcionando en un entorno productivo después de dos semanas de estabilización, de acuerdo a los requerimientos acordados.</li> <li>○ Capacitación al personal de Administración sobre el sistema de Facturación a brindarse a grupos de 15 personas, cuatro sesiones en total, a dictarse de 8 a 18 hs en casa central antes de la etapa de Prueba del sistema. La capacitación será exclusivamente teórica.</li> <li>○ Entrega de una copia del manual del sistema original en idioma inglés.</li> <li>○ Entrega de dos copias del manual de operación del sistema adaptado en el proyecto, en idioma español.</li> <li>○ Soporte remoto a ser brindado por un técnico en el horario de 8 hs a 22 hs, durante un período de dos meses desde la instalación del sistema y hasta un máximo de 200 horas.</li> <li>○ Instalación de una red Ethernet en las Oficinas de Talcahuano, consistente en cableado estructurado, Rack cerrado, puntos de red junto a cada escritorio (15 en total), .....</li> </ul>

#### No tener definido cómo se acepta el proyecto

DESCRIPCIÓN
El servicio o producto a entregar se puede especificar claramente, pero si no están pactados y escritos los criterios de aceptación es posible que el cliente rechace la recepción. Un proyecto Informático de envergadura, especialmente si se trata de Integración de Sistemas, tiene como objetivo producir una solución a ser entregada al cliente. Esta solución es un conjunto de "entregables" de Hardware, Software y Servicios que dependen de cada proyecto en particular. Cada una de estas componentes posee especificaciones propias tanto en qué es lo que deben hacer, cómo deben hacerlo y qué tan bien deben hacerlo. Estos "entregables" son producidos a todo lo largo del proyecto y pueden ser de diferente naturaleza, por citar algunos: <ul style="list-style-type: none"> <li>● Componentes de Hardware instalados y funcionando</li> <li>● Componentes de Hardware que serán instalados por el cliente</li> <li>● Dictado de cursos</li> <li>● Provisión una red de comunicaciones</li> <li>● Provisión de personal para operación del Centro de Cómputos</li> <li>● Sistema aplicativo funcionando en dependencias del cliente</li> <li>● Software de Base instalado en el Computador Central del Cliente</li> </ul> Todos y cada uno de los "entregables" del proyecto deben estar establecidos con anterioridad al inicio del mismo. En ese momento, para cada uno de los "entregables" debe también establecerse un Criterio de Aceptación, esto es, una descripción de las condiciones o hechos que deben cumplirse para que el cliente dé por aceptado en forma satisfactoria el "entregable" en cuestión.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Tiempo: El proyecto se demora hasta que se logra lo requerido por el cliente en el momento de la aceptación</li> <li>○ Alcance: Se acepta el producto pero el cliente no queda satisfecho con lo recibido.</li> </ul>
PREVENCIÓN
Especificar claramente en el Contrato de Servicios los criterios de aceptación para cada servicio o producto
Ejemplo
<p><b>Criterios de Aceptación</b></p> <ul style="list-style-type: none"> <li>○ Las librerías contenido los Programas Cobol serán aceptadas cuando se entreguen acompañadas de la compilación de cada programa y código de condición no mayor a 4.</li> <li>○ Las pruebas serán aceptadas cuando hayan transcurrido las dos semanas de prueba y no existan errores pendientes de arreglo. No se considerarán errores los cambios y mejoras solicitadas.</li> <li>○ Se considerará cumplido el servicio de soporte cuando hayan transcurrido dos meses desde la puesta en producción del sistema.</li> <li>○ Licencias de Programas: Se considerará entregada la licencia cuando se hayan ejecutado los procedimientos de verificación de instalación (IVP).</li> <li>○ Capacitación: Se considerará brindado el servicio cuando se hayan cumplido los días de dictado previstos para cada curso.</li> </ul>

- *Adaptación de Aplicaciones:* Se dará por aceptada la Adaptación de la Aplicación al haberse ejecutado en forma satisfactoria en el ambiente de desarrollo (en un porcentaje a convenir con el Cliente) un lote de prueba acordado entre el Cliente y el Integrador. En caso de que el Cliente no entregue o no acuerde con el Integrador dicho lote de prueba en los tiempos definidos en el Plan de Trabajo, el Integrador determinará y desarrollará dicho Lote de Prueba. La ejecución satisfactoria del mismo en el ambiente de desarrollo (en un porcentaje a determinar por el Integrador) será suficiente para dar por aceptada la adaptación de la aplicación. El Integrador notificará al Gerente de Proyecto del Cliente una vez concluida satisfactoriamente la prueba. El Cliente dispondrá de un plazo de 5 días hábiles para comunicar en forma fehaciente cualquier observación sobre el resultado de la prueba. Transcurrido este plazo sin mediar observaciones por parte del Cliente o una vez resueltas las mismas, se dará por cumplido el servicio.

#### No tener definido cuándo se entrega algo

DESCRIPCIÓN	
En todo contrato se determinar la fecha de entrega de un servicio o sistema, pero muchas veces se descomponen las actividades o la funcionalidad se entrega en fases y allí es donde no queda claro cuándo se entregará cada servicio o componente.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: El proyecto no se entrega en su totalidad en la fecha esperada y se demora el resto de la funcionalidad.</li> </ul>	
PREVENCIÓN	
Especificar claramente en el Contrato de Servicios cuándo se entregará cada fase o componente y cuándo se dispondrá de la totalidad del sistema / funcionalidad requerida.	
Ejemplos	
<b>Entregas parciales</b> <ul style="list-style-type: none"> <li>○ El módulo de facturación estará en producción el 15 de enero.</li> <li>○ La capacitación se brindará con la entrega del segundo módulo el 25 de marzo.</li> <li>○ Los manuales de uso se entregarán una semana antes de la puesta en producción del sistema.</li> </ul>	

#### No tener definidos niveles de Soporte

DESCRIPCIÓN	
La transferencia de conocimiento (Know How) en los proyectos informáticos es importante y se supone que en la transición la firma encargada del proyecto proveerá soporte y asistencia al cliente. Si dicho soporte no se especifica correctamente, es posible que el proveedor no esté dispuesto a brindar el nivel de soporte que el cliente puede requerir en algún momento determinado.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Costo: El proveedor cotizará el soporte adicional incrementando el costo del proyecto.</li> <li>○ Alcance: El cliente no termina recibiendo el nivel de servicio que esperaba.</li> </ul>	

#### PREVENCIÓN

Especificar claramente en el Contrato de Servicios todos los tipos de soporte que se brindarán cuantificando lo más posible cada uno.

#### Ejemplos

##### Soporte

- Se brindará soporte durante la ejecución de las pruebas de aceptación. Dicho soporte consistirá en el acompañamiento en el lugar de dos analistas experimentados para resolver todas las dudas que surjan, registrar los errores encontrados y ayudar a los usuarios en la utilización del nuevo sistema.
- Se brindará soporte post producción mediante el servicio de mesa de ayuda de Lunes a Viernes de 9 a 20hs. Este servicio consiste en una central de llamadas que derivará a los analistas experimentados todas las consultas recibidas, pudiendo atender hasta 5 consultas en forma simultánea.

#### No tener definida la capacitación

DESCRIPCIÓN	
La transferencia de conocimiento (Know How) en los proyectos informáticos es importante y se supone que en la transición la firma encargada del proyecto capacitará al la organización del cliente. Pero también se ve a la capacitación como una actividad del proceso de Gestión del Cambio que usualmente es responsabilidad del cliente. Por lo tanto es importante que se explice el tipo de capacitación que se brindará.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Costo: El proveedor cotizará por separado la capacitación solicitada en forma tardía, incrementando el costo del proyecto.</li> <li>○ Alcance: El cliente no aprovecha el producto que recibe pues su personal no está entrenado para usar el sistema.</li> </ul>	
PREVENCIÓN	
Especificar claramente en el Contrato de Servicios la capacitación que se debe brindar.	
Ejemplos	
<b>Capacitación</b> <ul style="list-style-type: none"> <li>○ Se capacitará al personal de la sucursal fuera de las horas laborales y en sus mismos puestos de trabajo. La capacitación consistirá en 30 ejercicios prácticos cubriendo el grueso de las transacciones que han sufrido modificación en las campos de importe...</li> <li>○ La capacitación consistirá en el entrenamiento de 5 analistas del cliente en la modalidad "Capacitar a los capacitadores" a fin de que ellos a su vez capaciten al resto del personal. Los cursos consistirán de 8 sesiones de 4 horas cada una, se entregará material de capacitación para ser reproducido por el cliente para las subsiguientes capacitaciones...</li> </ul>	

#### No cuantificar todo

DESCRIPCIÓN	
Los proyectos informáticos contienen Hardware, Software y Servicios. No hay duda sobre la necesidad de cuantificar el Hardware que se entregará, pero cuando se trata de Software y Servicios frecuentemente se entra en el terreno de las imprecisiones, especialmente cuando se habla de soporte, de mantenimiento, de errores, de calidad.	

EFEKTOS	
<ul style="list-style-type: none"> <li>○ Alcance: El cliente no recibe lo que esperaba</li> </ul>	
PREVENCIÓN	
<p>Especificar claramente en el Contrato de Servicios la cantidad de todo lo provisto y si un servicio no es cuantificable, establecer un acuerdo de nivel de servicio medible.</p>	
Ejemplos	
<ul style="list-style-type: none"> <li>○ Se corregirán sin cargo hasta 100 errores durante el primer año de uso del sistema.</li> <li>○ Se entregará el sistema funcionando y 1 backup en cartucho compatible con el equipamiento del cliente.</li> <li>○ Se brindarán 400 horas de soporte técnico en el lugar (on site).</li> <li>○ Se garantiza un tiempo de disponibilidad central del 99,97% (ver SLA uptime).</li> </ul>	
<b>No requerir ni disponer de Infraestructura de trabajo: espacio físico, sillas, escritorios, PC, teléfonos</b>	
DESCRIPCIÓN	
<p>No se especifica dónde el proveedor desarrollará el proyecto y si se indica, es común no alertar sobre las necesidades que tendrá el equipo de proyecto.</p>	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Costo: El cliente termina comprando mobiliario o equipamiento adicional</li> <li>○ Tiempo: Es posible que se retrase el proyecto hasta no disponer de la infraestructura necesaria.</li> </ul>	
PREVENCIÓN	
<p>Especificar claramente en el Contrato de Servicios la infraestructura necesaria para que se desarrolle un proyecto.</p>	
Ejemplos	
Responsabilidades del Cliente - Infraestructura	
<ul style="list-style-type: none"> <li>○ <i>El cliente deberá contar con espacio de oficina para el desarrollo del proyecto. Deberá proveer escritorios para 15 personas, 15 computadores personales con la configuración contratada para la sucursal piloto, 10 teléfonos con acceso a llamadas externas, sillas tipo ...., Una sala de reunión con una mesa y capacidad para 8 personas sentadas.</i></li> <li>○ <i>El cliente deberá proveer entornos de desarrollo y de prueba que tengan un tamaño equivalente al de 30% del entorno de producción actualmente utilizado.</i></li> </ul>	
<b>No contar con acceso fuera de horario normal de oficina</b>	
DESCRIPCIÓN	
<p>Usualmente los proyectos informáticos sufren contratiempos y es necesario trabajar fuera de las horas usuales y los fines de semana. Puede suceder que las instalaciones del cliente no tengan habilitado el trabajo fuera de horas normales</p>	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Costo: El cliente termina contratando fuera de presupuesto, servicios de seguridad para permitir el trabajo en horas no usuales o pagando horas extraordinarias al personal de seguridad existente.</li> </ul>	

EFEKTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: de no ser posible trabajar fuera de horario será muy difícil recuperar los atrasos, lo cual impactará la fecha de entrega comprometida.</li> </ul>	
PREVENCIÓN	
<p>Especificar claramente en el Contrato de Servicios la necesidad de realizar trabajos fuera de horario.</p>	
Ejemplos	
Responsabilidades del Cliente – Infraestructura	
<ul style="list-style-type: none"> <li>○ <i>El cliente deberá permitir la entrada y permanencia del personal del proyecto durante las 24 horas del día los 7 días de la semana.</i></li> </ul>	
<b>No especificar el costo de horas adicionales</b>	
DESCRIPCIÓN	
<p>Si los contratos cuantifican apropiadamente los servicios a proveer es posible que omitan qué sucede cuando se requieren servicios adicionales por sobre lo contratado. Existe un problema de disponibilidad de recursos del proveedor, pero más importante aún, del costo que estos servicios extra tendrán.</p>	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Costo: El cliente no contempla tener que pagar por el proyecto más allá de lo especificado en el contrato.</li> </ul>	
PREVENCIÓN	
<p>Especificar claramente en el Contrato de Servicios el costo de las horas adicionales.</p>	
Ejemplos	
<ul style="list-style-type: none"> <li>○ <i>El servicio de soporte se brindará hasta un total de 200 horas, en caso de que el cliente solicite más horas de soporte estas se cobrarán en forma adicional a 1 UF cada hora, esto es equivalente al 150% de la hora hombre de un analista experimentado.</i></li> </ul>	
<b>No contar con la propiedad del código o de los fuentes</b>	
DESCRIPCIÓN	
<p>En la legislación de muchos países se asume que quien contrata los servicios de desarrollo de software recibe por defecto la propiedad del código objeto o los programas fuentes. Si por algún motivo los derechos de propiedad o de las obras derivadas quedase en manos distintas a las del cliente, este tendrá una dependencia eterna con quien tenga la propiedad del software y este último podría ejercer prácticas monopólicas como imponer precios abusivos al mantenimiento o nuevas versiones del software en cuestión.</p>	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Costo: El cliente podría pagar precios abusivos por trabajos subsiguientes sobre el mismo Software.</li> <li>○ Alcance: La expectativa del cliente se ve vulnerada ya que este espera ser propietario de Software.</li> </ul>	
PREVENCIÓN	
<p>Es recomendable reconfirmar con una cláusula en el Contrato de Servicios de quién es la propiedad del software que se desarrolle o personalice en el proyecto.</p>	

**Ejemplos**

- Se establece que las partes dejan expresa constancia que la totalidad de los diseños, productos de software y documentación que se desarrolle con motivo de los servicios contratados, y que constituyen el objeto del presente contrato, son de exclusiva propiedad del CLIENTE. Asimismo, EL PROVEEDOR se obliga a que los contratos de trabajo que celebre con su personal incorporen esta obligación, firmando un anexo cláusula de propiedad.
- XXX transferirá la propiedad de los bienes entregados o de los servicios realizados una vez éstos hayan sido aceptados por el CLIENTE.
- Para el software suministrado, el CLIENTE adquiere la licencia de uso del mismo. Dicha licencia será no exclusiva, no transmisible, no sublicenciable con una duración de \_\_\_\_\_ años a partir de la fecha de entrega. El CLIENTE, sin el previo consentimiento de XXX:

**No especificar soporte post-producción**

DESCRIPCIÓN
Es común concentrar muchos esfuerzos en determinar todas las tareas y servicios que se deben proveer hasta que el proyecto se encuentre entregado y productivo al punto tal que se puede obviar el especificar el tipo de soporte que se brinda posteriormente. Este soporte se puede considerar una especie de Período de Garantía en el cual se deben resolver gratuitamente todas las incidencias que surjan y brindar soporte ante los problemas de uso que surjan.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Costo: El cliente podría tener que pagar montos adicionales por los servicios no especificados.</li> </ul>
PREVENCIÓN
Especificar claramente en el Contrato de Servicios el tipo de soporte post-producción que se brindará.
Ejemplos
<ul style="list-style-type: none"> <li>○ A partir de la aceptación del Cliente de la instalación del Software en un ambiente productivo, se brindará un servicio de soporte post implantación que consistirá en:           <ol style="list-style-type: none"> <li>1. La presencia durante horario laboral de un equipo de dos analistas expertos en el Software a fin de resolver dudas, diagnosticar fallas y resolver problemas simples.</li> <li>2. La corrección de errores que se pudieran detectar. Los errores serán resueltos según su severidad en los tiempos comprometidos en el Acuerdo de Nivel de Servicio (SLA) Anexo.</li> </ol> </li> </ul>

**No especificar mantenimiento**

DESCRIPCIÓN
En ocasiones no se hace mención alguna sobre servicios de mantenimiento posteriores al período de garantía. Esto puede generar falsas expectativas en el cliente.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Alcance: Decepción del cliente que puede pensar que los errores de funcionamiento serán solucionados por el proveedor en el futuro en forma gratuita.</li> </ul>

**PREVENCIÓN**

Se debe mencionar esta situación en la etapa de preventa y llegado el caso especificarlo en el Contrato de Servicios.

**Ejemplos**

- Quedan incluidas todas las actuaciones del PROVEEDOR remotas y on-site (desplazamientos de los técnicos del PROVEEDOR) necesarias para la solución de las averías cumpliendo con los tiempos de resolución pactados. Soporte software: quedan incluidas las actualizaciones de software necesarias para corregir cualquier mal funcionamiento de los equipos. A su vez se actualizarán los sistemas operativos y aplicaciones de los equipos con las nuevas versiones que libra el fabricante y para añadir nuevas funcionalidades disponibles. En caso que la implantación de las nuevas versiones de software requieran la instalación de un nuevo hardware, éste será a cargo del CLIENTE. La fecha de introducción de las nuevas versiones de software será pactada entre el CLIENTE y el PROVEEDOR.
- El servicio de mantenimiento incluirá, en todo caso, la corrección de errores y el soporte técnico para diagnosticar fallas, excluyéndose la actualización e implantación de nuevas versiones que el PROVEEDOR pudiera generar en el futuro.

**Realizar cotizaciones incompletas**

DESCRIPCIÓN
El proveedor puede llegar a cotizar un servicio que en realidad no representa la totalidad del servicio que espera recibir el cliente. Esto puede originarse por problemas de comunicación o bien porque el proveedor no tiene capacidad de brindar la totalidad del servicio.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Costo: El proyecto termina costando mucho más de lo presupuestado.</li> <li>○ Alcance: Decepción del cliente que no recibe la totalidad del servicio.</li> </ul>
PREVENCIÓN
Se debe cotizar en forma completa o especificar claramente lo que no se brindará.
Ejemplos
<ul style="list-style-type: none"> <li>○ El servicio consiste en la instalación y puesta a punto de una red de computadoras consistente en un servidor, 2 impresoras y 10 computadores personales. El servicio no incluye el cableado estructurado que estará a cargo del cliente.</li> </ul>

**No especificar cómo tratar los temas relativos a la Confidencialidad**

DESCRIPCIÓN
El personal del proveedor tendrá durante el proyecto una fuerte interacción con el personal, documentos y datos del cliente. Es posible que la información que alguna de las personas obtenga pueda ser transmitida intencionalmente o no a terceros causando una diversidad de perjuicios.
EFFECTOS
<ul style="list-style-type: none"> <li>○ Costo: El negocio del cliente se puede ver afectado por la fuga de información con consecuencias económicas.</li> </ul>
PREVENCIÓN

Se debe establecer en el contrato de servicios las necesidades o limitaciones relacionadas con la confidencialidad.

#### Ejemplos

- *EL PROVEEDOR, sus directores, consejeros, funcionarios y el personal contratado por éste, se obligan a no divulgar a tercera personas naturales y/o jurídicas públicas o privadas, firmas o instituciones y/o hacer uso personal o de la empresa, de cualquier información confidencial del CLIENTE y/o de cualquiera de sus clientes. El término "Información Confidencial" comprende cualquier clase de información a que tenga acceso EL PROVEEDOR en virtud del presente contrato en relación al CLIENTE. Además comprende toda información desarrollada o no por el CLIENTE, referente a las actividades del CLIENTE o sus empresas filiales coligadas o sus clientes, así como aquellas que confieren al CLIENTE una ventaja respecto a sus competidores que no poseen dicha información; incluyendo, entre otras, actividades de investigación desarrolladas por el CLIENTE, especificaciones técnicas, listados, métodos, sistemas, información sobre el personal, datos financieros actuales y estimados. Asimismo, EL PROVEEDOR se obliga a que los contratos de trabajo que celebre con su personal contengan la referida obligación de confidencialidad, firmando un anexo cláusula de confidencialidad. Las obligaciones de confidencialidad emanadas de esta sección continuarán vigentes aun después de la fecha de terminación del Contrato.*
- *Ambas partes se comprometen a guardar secreto sobre cualquier hecho, noticia o dato del que tengan conocimiento en virtud de la conclusión, ejecución y/o resolución del presente contrato, incluidos los términos del mismo, exceptuando los supuestos en que sus empleados o agentes tengan legítima necesidad de conocer tales términos para el debido cumplimiento de las obligaciones que emanen del mismo, y en caso de resolución por cualquier causa.*

#### No considerar aspectos impositivos y tributarios del proyecto

##### DESCRIPCIÓN

Es común que los aspectos impositivos y tributarios sean una de las últimas prioridades en los proveedores de servicio de Software. Por lo tanto no existirán cláusulas ni advertencias sobre impuestos que el cliente pudiera tener que pagar sobre el servicio recibido o las licencias de software que componen el proyecto.

##### EFFECTOS

- Costo: El proyecto termina costando mucho más de lo presupuestado.

##### PREVENCIÓN

Se debe conversar e investigar durante la preventa los aspectos impositivos del proyecto y de ser posible incluir las respectivas responsabilidades en el contrato de servicios.

#### Ejemplos

- *Las partes declaran que en atención a la naturaleza de los servicios que se prestarán en virtud del presente Contrato de desarrollo, adecuación e implementación de software, y con sujeción a lo instruido por el Servicio de Impuesto Internos mediante los Oficios \_\_\_\_\_; todos ellos en relación con la aplicación de lo dispuesto en el artículo \_\_\_\_\_ de la Ley sobre Impuesto a la Renta y en el artículo \_\_\_\_\_ de la ley sobre Impuesto a las Ventas y Servicios; tales servicios y prestaciones que el CLIENTE ha convenido en este instrumento con EL PROVEEDOR no se encuentran gravadas con el Impuesto al Valor Agregado (IVA), y por ende no procederá su recargo en la correspondiente factura.*

#### No determinar qué hacer ante problemas con personas que participan en el proyecto

##### DESCRIPCIÓN

Dentro del equipo de profesionales que el proveedor asigne al proyecto, puede haber personas que no satisfagan las expectativas del cliente. Muchas veces la no aceptación de determinadas personas radica en cuestiones de personalidad y actitud. Estos temas generalmente generan bastante controversia y dependiendo de los términos contractuales, será decisión del proveedor si reemplaza a estos profesionales o no.

##### EFFECTOS

- Tiempo: El proyecto se puede retrasar si el o los reemplazos tienen una curva de aprendizaje para ponerse al día con el proyecto.

##### PREVENCIÓN

Se debe especificar en el contrato de servicios que el cliente se reserva el derecho de solicitar el cambio de personal ante problemas de comportamiento o actitud y que el proveedor deberá reemplazar a el o los profesionales en cuestión en un plazo dado y sin impactar las fechas del proyecto.

#### Ejemplos

- *Es condición determinante de la celebración, ejecución y duración del presente Contrato, que EL PROVEEDOR informe oportunamente todos los antecedentes del personal que participe como parte del equipo de proyecto. El CLIENTE tiene derecho a pedir el reemplazo de algunos miembros del equipo de proyecto del PROVEEDOR hasta un 20% de la totalidad del personal aportado por el PROVEEDOR, no siendo necesario contar con la aceptación por parte del PROVEEDOR de los argumentos que esgrima el CLIENTE para solicitar los respectivos cambios.*

#### Composición del equipo de trabajo

##### DESCRIPCIÓN

Durante el periodo de preventa los proveedores hacen participar a personal con perfil de ventas o consultores de larga experiencia, pero al comenzar la ejecución del proyecto el personal del proveedor puede tener un perfil no tan convincente. En los proyectos modalidad "Tiempo y Materiales", se pueden objetar la composición de los equipos mientras que en los proyectos modalidad "Llave en Mano" donde el compromiso está en entregar algo en determinado plazo y precio, el proveedor determina con cuántos y quiénes satisface el compromiso y por lo tanto la conformación del equipo de proyecto será entera responsabilidad del proveedor.

##### EFFECTOS

- Tiempo: Un equipo de proyecto no idóneo difícilmente pueda culminar el proyecto en tiempo.

##### PREVENCIÓN

En los proyectos modalidad "Tiempo y Materiales" se deberá solicitar que se especifique en el contrato de servicios o algún anexo la composición de los equipos de trabajo, citando roles y nivel de experiencia.

#### Ejemplos

- *El equipo de Proyecto provisto por el proveedor consistirá de:*
  - *Un Jefe de Proyecto Senior con experiencia en por lo menos 2 implementaciones similares del sistema X*
  - *Dos analistas Senior con 10 años de experiencia*
  - *Cuatro analistas programadores con 5 años de experiencia*

**Rotación del equipo de trabajo**

DESCRIPCIÓN	
Un proyecto puede comenzar con un buen equipo de trabajo y es posible que al proveedor le surja otro proyecto y decida reconfigurar el equipo de trabajo. La rotación de personal resultante puede impactar negativamente la evolución del proyecto.	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: Un cambio en el equipo de proyecto resulta por lo general en una demora en el proyecto.</li> </ul>	
PREVENCIÓN	
En los proyectos modalidad "Tiempo y Materiales" se debe especificar en el contrato de servicio que deben evitarse los cambios en la conformación del equipo de trabajo.	
Ejemplos	
<ul style="list-style-type: none"> <li>○ Una vez conformado el equipo de trabajo del proyecto provisto por EL PROVEEDOR, los cambios, rotación y reasignaciones de personal determinados por EL PROVEEDOR deberán realizarse con la aprobación del CLIENTE.</li> </ul>	

**No prevenir los fracasos**

DESCRIPCIÓN	
Los proyectos informáticos pueden implicar inversiones muy significativas. Si el proveedor de servicios falla, las inversiones conexas pueden quedar improductivas por tiempo indeterminado. El proveedor puede fallar por una cantidad de motivos, desde la inviabilidad del proyecto propuesto hasta fallas metodológicas que serán mencionadas más adelante. Si el proveedor de servicios quiebra en medio del proyecto, tendremos los mismos resultados.	
EFEKTOS	
<ul style="list-style-type: none"> <li>○ Costo: Un fracaso del proyecto implica pérdidas económicas importantes.</li> </ul>	
PREVENCIÓN	
Se recomienda contar con cláusulas de salida ante la cancelación del proyecto y seguros de caución que cubran económicamente el fracaso del proveedor. Para cubrir este tipo de situaciones se puede firmar en forma conjunta un contrato de ESCROW, con el cual se puede especificar que todo el trabajo que se vaya realizando se almacene en dependencias seguras de un tercero, y por ejemplo ante quiebre del proveedor o cancelación del contrato, el Cliente tenga acceso a dicho material. En los casos en que después de finalizado el Proyecto el cliente no tiene copia de los programas fuente, es recomendable firmar un contrato ESCROW que le dé al Cliente acceso a dichos fuentes en el caso que el proveedor quiebre; inclusive se puede solicitar tener acceso a la lista de personas (incluyendo teléfonos de contacto) que han trabajado en el proveedor y que tienen conocimiento del Software en cuestión.	
Ejemplos	
<ul style="list-style-type: none"> <li>○ Las partes podrán poner fin al presente contrato:           <ol style="list-style-type: none"> <li>1. En caso de incumplimiento del presente contrato, en cualquiera de sus puntos.</li> <li>2. Ante el no cumplimiento por parte de EL PROVEEDOR de entregar los productos de software detallados en cada Anexo, en las condiciones de calidad y plazos exigidos por el CLIENTE.</li> </ol> </li> </ul>	

**PROBLEMAS GENERALES**

Las siguientes situaciones se dan con mayor o menor frecuencia, pero de ocurrir alguna, es altamente probable que generen problemas. Por este motivo se ha denominado a este conjunto de situaciones, Problemas Generales. Estos problemas no se resuelven a nivel contractual sino que son cuestiones que se deben considerar en el momento de preventa y análisis de factibilidad:

1. Hacer proyectos muy largos
2. Pretender implementar demasiados sistemas y funcionalidad juntos
3. No tener entornos de prueba
4. No realizar gestión del cambio
5. Tener conocimiento concentrado en pocas personas
6. No tener / sacar backups
7. Guardar la información en disco duro de las máquinas de los que desarrollan
8. Tener problemas de idioma
9. No realizar capacity planning
10. No prever vacaciones, enfermedades, embarazos, tiempo de traslado, curva de aprendizaje
11. Rompecabezas tecnológico
12. Llave en mano no significa sin colaboración
13. Usar la última tecnología disponible
14. Sobreventa
15. Construir el sistema equivocado
16. Construir un sistema que nadie necesita
17. Construir un sistema en base a requerimientos de un usuario que no es el usuario real
18. No tener nadie que apruebe o firme
19. No identificar al sponsor ni a otros involucrados
20. Desconocer el proyecto de la competencia (usualmente el no elegido)
21. No integrar (torre de babel)
22. Llave en mano versus tiempo y materiales
23. Se puede programar en una sola Terminal
24. Importar soluciones, la necesidad de "localizar"
25. Enlatados: ajustar mi proceso en vez del software
26. Las modas de Management y de la tecnología
27. Sistemas paramétricos y paradigmas distintos
28. Arquitectura de la solución
29. Fanáticos de una tecnología

30. Desventajas de un proveedor de demasiado peso  
 31. Tecnologías robustas y confiables versus tecnologías ágiles  
 32. Mantenibilidad de la aplicación  
 33. Limpieza de Datos  
 34. No tener lecciones aprendidas  
 35. Construir o comprar  
 36. No contar con recuperación ante desastres (*Disaster Recovery Planning*)  
 37. No contar con Infraestructura de comunicaciones  
 38. Proyectos multipaís, multiorario  
 39. Terminología común en contrato y proyecto  
 40. Bajo nivel de usuarios  
 41. Mala Comunicación

#### DESCRIPCIÓN DE LOS PROBLEMAS GENERALES

##### Hacer proyectos muy largos

DESCRIPCIÓN
Es común que los proyectos informáticos avancen más lentamente que las necesidades de negocio, por lo tanto en la medida que el proyecto que preparamos sea demasiado ambicioso es posible que demore muchos meses, si no años, y por lo tanto cuando esté listo ya no le sirva al cliente. Otra alternativa es que se vayan introduciendo cambios constantemente y el proyecto se vuelva ingobernable.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Costo: como máximo el costo total del proyecto será una pérdida, como mínimo no se obtendrá el beneficio de utilizar el sistema pues cuando esté listo se deberá actualizar para satisfacer las demandas vigentes.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Parcializar los proyectos, dividir la funcionalidad en etapas y fases que faciliten la implantación</li> </ul>

##### Pretender implementar demasiados sistemas y funcionalidad juntos

DESCRIPCIÓN
En cada proyecto, la implementación de cada sistema lleva consigo un cierto riesgo. Si se concentran varias implantaciones de sistemas los riesgos se multiplican.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Tiempo: es altamente probable que los sistemas no estén productivos en la fecha comprometida.</li> <li>○ Alcance: la calidad se puede ver seriamente impactada y se puede necesitar de mucho tiempo hasta estabilizar todos los sistemas implantados en forma conjunta.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Distribuir las implantaciones evitando a su vez fechas críticas como fines de mes.</li> </ul>

##### No tener entornos de prueba

DESCRIPCIÓN
Si no se dispone de entornos de prueba se corre el riesgo de mezclar datos de producción con datos de prueba o afectar con los sistemas en prueba los datos de producción.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Plazo y Alcance: la falta de un entorno de prueba primero demorará el proyecto hasta que se consiga un entorno adecuado. El no tener un entorno adecuado afectará la efectividad de la prueba y por ende la calidad del producto.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Especificar en los anexos técnicos la necesidad de contar con un entorno de prueba. Ver la posibilidad de arrendar espacio temporalmente.</li> </ul>

##### No realizar gestión del cambio

DESCRIPCIÓN
Todo sistema trae aparejados cambios en la manera de trabajar en la organización del cliente. Se denomina Gestión del Cambio al proyecto paralelo que el cliente debe realizar para crear o ajustar sus procesos administrativos, capacitar al personal, definir nuevos formularios, etc.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Alcance: el sistema no será exitoso en su implantación pues la organización no estará preparada para utilizarlo.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Desarrollar las tareas de Gestión del Cambio y controlar dicho proyecto en forma conjunta con el Proyecto Informático.</li> </ul>

##### Tener conocimiento concentrado en pocas personas

DESCRIPCIÓN
Tanto de parte del proveedor como del cliente suele suceder que el conocimiento del proyecto y nuevo sistema queda concentrado en pocas personas.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Tiempo: Si una persona clave del proveedor deja de participar se generarán demoras.</li> <li>○ Alcance: Si una persona clave del cliente deja la empresa cerca del momento de implantar el sistema (antes o después).</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Distribuir el conocimiento en varias personas. Formar a reemplazos.</li> </ul>

##### No tener o sacar backups

DESCRIPCIÓN
Durante el proyecto es importante ir generando copias de resguardo de todo el trabajo que se vaya realizando.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Tiempo: la perdida de documentos, archivos, librerías o programas genera retrasos</li> </ul>

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>Tomar backup del entorno de desarrollo y de los documentos de gestión y diseño del proyecto.</li> </ul>	

**Guardar la información en disco duro de las máquinas de los que desarrollan**

DESCRIPCIÓN	
Un error común es ir guardando todo el trabajo individual en el disco duro de la máquina de cada desarrollador.	

EFFECTOS	
<ul style="list-style-type: none"> <li>Tiempo: la pérdida de documentos, archivos, librerías o programas genera retrasos.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>Utilizar una estructura de directorio en un disco común o sea residente en un servidor. Tomar backup de dicho servidor diaria o semanalmente.</li> </ul>	

**Tener problemas de idioma**

DESCRIPCIÓN	
En los proyectos informáticos en forma frecuente se presenta la barrera del idioma, ya sea porque los consultores de determinado producto solo hablan inglés y descubrimos eso el día que arriban, o porque los manuales del producto original se encuentran en alemán.	

EFFECTOS	
<ul style="list-style-type: none"> <li>Costos: La contratación de traducción simultánea o de traducción de manuales incrementa los costos del proyecto.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>Tener en cuenta este aspecto con consultores, productos o documentación que provengan del exterior.</li> </ul>	

DESCRIPCIÓN	
El capacity planning (o planear la capacidad) consiste en determinar en una etapa temprana los consumos que tendrá el sistema que se desarrollará y por lo tanto la capacidad que deberá tener en la infraestructura al momento de instalar el sistema. Por capacidad nos referimos a cantidad de espacio en disco, velocidad de acceso a datos almacenados, anchos de banda, velocidad de proceso. También se incluyen estudios de volumetría y crecimiento de datos en base a la historia.	

EFFECTOS	
<ul style="list-style-type: none"> <li>Costos: se incrementan los costos al tener que invertir en equipamiento y no tener esto presupuestado.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>Realizar estimaciones de capacidad, siempre.</li> </ul>	

DESCRIPCIÓN	
No prever vacaciones, enfermedades, embarazos, tiempo de traslado	

Los ejercicios de planificación que se realizan dentro de los proyectos informáticos, suelen ser bastante optimistas y raramente toman en cuenta la fecha en que realmente podrían comenzar, el tiempo que podrían demorar y las situaciones que normalmente podrían ocurrir en dicho período, tales como períodos de vacaciones, enfermedades de miembros del equipo de proyecto con licencias prolongadas, licencias por maternidad, o simplemente los tiempos de traslado cuando el proyecto se desarrolla en varios sitios simultáneamente y es necesario que miembros del equipo de proyecto se trasladen frecuentemente.

EFFECTOS	
<ul style="list-style-type: none"> <li>Plazos: seguramente se demorará el proyecto si no están previstos los reemplazos de personal o los tiempos improductivos de los trasladados.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>En el período de preventa se le debe preguntar al proveedor si ha planificado las actividades contemplando las situaciones planteadas.</li> </ul>	

**Rompecabezas tecnológico**

DESCRIPCIÓN	
Muchas soluciones tecnológicas implican combinar distintos elementos (Hardware y Software) de distintos fabricantes. Las especificaciones de cada producto describirán el nivel de compatibilidad de cada componente pero frecuentemente en la práctica surgen incompatibilidades que son difíciles de resolver. El principal problema radica en determinar dónde está el problema pues cada fabricante o representante indicará que el funcionamiento de su equipo o sistema es correcto y que la falla está en otro componente de un tercero.	

EFFECTOS	
<ul style="list-style-type: none"> <li>Costos: Frecuentemente se deberán buscar alternativas que resultan más costosas.</li> <li>Tiempo: La resolución de este tipo de problemas generalmente compromete la fecha de entrega final.</li> <li>Alcance: Es posible que no se llegue resolver el problema en forma transparente y por lo tanto se perderá parte de la funcionalidad que esperaba el cliente.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>En el momento de preventa es necesario preguntar si la combinación de los componentes se ha realizado antes en otros proyectos y cuáles fueron los resultados.</li> <li>Si es una combinación única o nueva, utilizar un proveedor como integrador de sistemas y que recaiga en él la responsabilidad de lograr el éxito de combinar los componentes propuestos.</li> </ul>	

**Llave en mano no significa sin colaboración**

DESCRIPCIÓN	
Cuando se firman contratos de servicio llave en mano es común que el personal del cliente entienda que no deben hacer nada pues de todo se encargará el proveedor.	

EFFECTOS	
<ul style="list-style-type: none"> <li>Plazos: se demorará el proyecto sin la participación de los empleados del cliente.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>Presentar el proyecto al equipo del cliente y hacer hincapié en la estrecha colaboración que deberá existir entre el equipo del proveedor y del cliente. Se recomienda mencionar las tareas que deberá realizar el personal del cliente.</li> </ul>	

**Usar la última tecnología disponible**

DESCRIPCIÓN	
EFFECTOS	RECOMENDACIÓN
En oportunidades dentro de la acción de venta se empuja al cliente a elegir la alternativa de proyecto que implica utilizar lo más novedoso tecnológicamente hablando. No está de más decir que utilizar la última tecnología implica serios riesgos, pues no todo está probado ni todo funciona de acuerdo a las especificaciones.	
<ul style="list-style-type: none"> <li>○ Tiempo: se demora el proyecto por tener que resolver problemas sin que existan experiencias similares en el mercado.</li> <li>○ Alcance: quizás no toda la funcionalidad anunciada con la nueva tecnología está disponible según se esperaba.</li> </ul>	<ul style="list-style-type: none"> <li>○ Utilizar tecnologías que estén probadas y estabilizadas. Es recomendable esperar un año con una tecnología o producto que recién sale al mercado.</li> </ul>
<b>Sobreventa</b>	

DESCRIPCIÓN	
EFFECTOS	RECOMENDACIÓN
Existen proveedores que se aprovechan del desconocimiento de los clientes y exageran sobre las capacidades de determinado sistema o producto informático. En ocasiones olvidan mencionar los puntos negativos o las implicancias de implantar determinada solución tecnológica.	
<ul style="list-style-type: none"> <li>○ Alcance: generalmente el cliente termina decepcionado pues no obtiene lo que esperaba recibir según las expectativas que la acción de preventa le generó.</li> </ul>	<ul style="list-style-type: none"> <li>○ Buscar una segunda opinión. Obtener cotizaciones de distintos proveedores. Exigir conocer y comprobar casos de éxito anteriores en proyectos similares.</li> </ul>

**Construir el sistema equivocado**

DESCRIPCIÓN	
EFFECTOS	RECOMENDACIÓN
Esta situación es muy triste y excepcional, pero puede ocurrir que el cliente crea que necesita un sistema, el proveedor se lo construye y al implantarlo el cliente no resuelve el problema que tenía. Cada día es más frecuente que cuando el cliente expresa su necesidad lo realice en términos de sistemas en vez de describir los problemas que necesita solucionar.	
<ul style="list-style-type: none"> <li>○ Alcance: El cliente no logra solucionar sus problemas con la funcionalidad implementada.</li> </ul>	<ul style="list-style-type: none"> <li>○ El cliente debe siempre explicitar los problemas que necesita resolver y asegurarse que queden relacionados con la solución que aporta el proyecto.</li> </ul>

**Construir un sistema que nadie necesita**

DESCRIPCIÓN	
EFFECTOS	RECOMENDACIÓN
Esta situación es común cuando a) el cliente forma parte de una corporación multinacional y por órdenes de la casa matriz se debe implementar determinado sistema. b) El dueño o Gerente General de una compañía decide copiar a un competidor o a otra compañía exitosa y decide instalar el sistema que aparentemente transformó a las otras compañías. Ante esta situaciones, lo que generalmente ocurre es que quedan dos sistemas funcionando (el anterior y el nuevo) o simplemente nadie llega a utilizar el nuevo sistema.	
<ul style="list-style-type: none"> <li>○ Alcance: El cliente no logra nada pues el sistema no se usa.</li> </ul>	<ul style="list-style-type: none"> <li>○ Para el caso a) como el sistema se tiene que implantar por obligación es necesario sacarle el mayor provecho, para ello se debe primero involucrar a los futuros usuarios y asegurarse que ellos se involucren de lleno antes de comenzar el proyecto inclusive en los casos en que ellos deban resignar un mejor sistema para adoptar el corporativo.</li> <li>○ Para el caso b) El proveedor debería hacer entrar en razón a quien haya determinado la implantación del sistema y explicarle las diferencias con las compañías que fueron exitosas con el sistema que se pretende adquirir y destacarle los pasos previos que habría que realizar para llegar a estar en la misma situación que esas otras compañías.</li> </ul>

**Construir un sistema en base a requerimientos de un usuario que no es el usuario real**

DESCRIPCIÓN	
EFFECTOS	RECOMENDACIÓN
Esta situación es muy común, al proveedor lo convoca una compañía, le señala la necesidad que tiene y le presenta a un usuario. Este usuario es muy diligente, describe toda la funcionalidad requerida, participa en el proyecto pero cuando el proyecto se pone en producción aparecen quejas y reclamos de otras áreas de la compañía. Una manera de comprender mejor esta situación es ver un ejemplo. Un usuario de Recursos Humanos puede definir un sistema de información de personal y definir unos magníficos reportes que utilizará su función, pero le presta muy poca atención al diseño de las pantallas que deberán ser utilizadas por los cientos de empleados de la compañía. Como resultado cuando los cientos de empleados comienzan a utilizar el sistema ven infinidad de deficiencias que el usuario de recursos humanos jamás habría detectado.	
<ul style="list-style-type: none"> <li>○ Alcance: la primera versión del sistema no alcanza a cubrir satisfactoriamente la funcionalidad que se necesita.</li> </ul>	<ul style="list-style-type: none"> <li>○ El proveedor deberá identificar todos los usuarios del sistema e involucrarlos a todos por igual.</li> </ul>

**No tener a nadie que apruebe o firme**

DESCRIPCIÓN	
En un proyecto existen innumerables instancias en las cuales se requiere de una aprobación formal e inclusiva de la firma de un funcionario. Podemos estar simplemente ante la aceptación de una minuta de reunión o la recepción de un producto intermedio. Esto sucede en compañías en las que el poder no está delegado o reparticiones públicas en donde los empleados son reacios a firmar por miedo a ser sumariados por futuras administraciones contrarias al proyecto.	

EFFECTOS
<ul style="list-style-type: none"> <li>• Tiempo: Si por metodología el proveedor no avanza sin tener las aprobaciones formales y estas demoran, terminarán impactando la fecha de entrega final.</li> <li>• Alcance: si se avanza con el proyecto sin haber reconfirmado requerimientos, diseño o pruebas, se puede llegar a entregar una solución que no sea lo que realmente se necesitaba.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>• Si por metodología el proveedor necesita las aprobaciones formales, deberá planificar más tiempo entre actividades. Asimismo se recomienda incluir un plazo de aprobación dentro de las cláusulas del contrato de servicio, indicando que si dichos plazos no se cumplen serán responsabilidad del cliente las demoras y costos adicionales.</li> </ul>

**No identificar al sponsor ni a otros involucrados**

DESCRIPCIÓN
Todo proyecto debe tener un sponsor, alguien que lo patrocine, alguien que está interesado en que el proyecto se realice y que está dispuesto a pagar por él. En un proyecto también existen otras personas y grupos involucrados. Es posible que ninguna de estas personas sean las que identificamos como importantes desde el punto de vista contractual. En la medida que no tengamos en cuenta a estas personas en nuestras comunicaciones sobre el proyecto o que no atendamos su opinión durante la ejecución del proyecto, estas personas podrían perder el interés en el proyecto y nosotros encontrarnos de pronto sin sustento político para el avance del proyecto.
EFFECTOS
<ul style="list-style-type: none"> <li>• Tiempo: Un proyecto sin sponsor es muy probable que nunca finalice.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>• En el momento de preventa se debe identificar al sponsor y durante la ejecución del proyecto el proveedor se debe asegurar de mantenerlo informado sobre la marcha del proyecto y atender sus comentarios.</li> <li>• También hay que identificar a otras personas claves dentro de la organización del cliente que pueden llegar a tener un efecto directo sobre el proyecto. Comúnmente estas personas son el Gerente de Finanzas, el Gerente de Sistemas, el Gerente de Organización.</li> </ul>

**Desconocer el proyecto de la competencia (usualmente el no elegido)**

DESCRIPCIÓN
A veces ocurre que dentro de una misma empresa hay dos o más grupos pugnando porque determinada problemática se resuelva con un proyecto/solución de su preferencia. Obviamente existirá un proceso de decisión y será solo uno el proyecto elegido. Cuando el equipo de Proyecto comienza a desarrollar sus tareas ignora por lo general que área favorecía otra solución. Esto es clave conocerlo lo antes posible pues es altamente probable que el área en cuestión sea la menos participativa y hasta incluso lleve a boicotear el proyecto.
EFFECTOS
<ul style="list-style-type: none"> <li>• Plazos: generalmente los proyectos se atrasan como producto de la acción de áreas opositoras al proyecto.</li> </ul>
RECOMENDACIÓN

- El proveedor deberá utilizar sus mejores habilidades como para involucrar a las áreas opositoras y convencerlas que el proyecto es su proyecto.

**No integrar (torre de babel)**

DESCRIPCIÓN
Como se mencionó anteriormente, muchos proyectos informáticos consisten en la integración de diversos componentes (hardware, software, servicios); se recomendó anteriormente seleccionar a un proveedor para que realice y se haga responsable de la integración. Hay ocasiones en que el cliente se hace cargo de la integración o peor aún, el cliente no es consciente de que se necesita integración alguna. La consecuencia es que todos los proveedores harán lo suyo, proveerán su servicio, construirán sistemas, pero no hablarán entre sí y si lo hacen no se entenderán porque cada uno tendrá distintos requerimientos y objetivos.
EFFECTOS

- Tiempo: el proyecto difícilmente termine, obviamente generando efecto sobre el costo y el alcance.

RECOMENDACIÓN
<ul style="list-style-type: none"> <li>• Ser consciente que la integración es necesaria y que es un valor agregado muy alto. Jamás el cliente debería hacerse cargo de la integración.</li> </ul>

**Llave en mano versus tiempo y materiales**

DESCRIPCIÓN
En otra sección hemos explicado estas dos modalidades de proyecto. El problema surge cuando a) tenemos un proyecto de integración de sistemas y el proveedor ha firmado un contrato llave en mano y a su vez el proveedor ha subcontratado con otros proveedores en modalidad tiempo y materiales. b) el cliente tiene un proyecto a plazo fijo y costo fijo y subcontrata tiempo y materiales. En el caso a) seguramente el proveedor va a quebrar y al cliente no le interesa quedarse sin proveedor durante el proyecto. El motivo por el que va a quebrar es fácil de ver con un ejemplo: supongamos que un contratista se compromete a pintar un edificio en un mes, calcula cuánto tiene que pagar de mano de obra durante un mes para pintar el edificio, pero comete la torpeza de subcontratar con otra firma el servicio de pintores de muro pero por hora. Es altamente probable que los pintores de muro tarden muchísimo más tiempo que el planificado y en la medida que dupliquen o tripliquen el tiempo planificado, los costos del proyecto serán demasiado altos para las espaldas del contratista principal. b) en ese caso es quien no puede cumplir con el plazo y el costo es el cliente.
EFFECTOS

- Tiempo: Los proyectos demorarán más de lo previsto.

RECOMENDACIÓN
<ul style="list-style-type: none"> <li>• Alertar sobre los peligros de esta ecuación.</li> </ul>

**Se puede programar en una sola Terminal**

DESCRIPCIÓN
Esta situación es muy poco frecuente. Se da cuando se utiliza alguna solución propietaria que construye sobre algún tipo de núcleo. Por más planificación y recursos que dispongamos, la imposibilidad de construir la solución en más de una Terminal o PC genera un cuello de botella fatal.
EFFECTOS

- Tiempo: El proyecto indefectiblemente no será entregado en el plazo acordado.

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Cuando se trate de una solución que se construya utilizando una herramienta poco conocida no está de más preguntar si la herramienta permite múltiple programación durante la construcción de la solución.</li> </ul>	

**Importar soluciones, la necesidad de "localizar"**

DESCRIPCIÓN	
<p>Muchos proyectos informáticos giran alrededor de la implantación de un software importado (a veces denominado paquete) y al momento de planificar el proyecto se ignora o subestima el esfuerzo que será necesario para "localizar" el sistema. Localizar es un término utilizado para englobar las adaptaciones que se le realizan a un producto para que cumpla con la normativa de un país. Ejemplos de esto son impuestos (IVA), identificador de personas (RUN/RUT en vez de Social Security Number), símbolo de moneda local, formato de fecha y hasta cambio de palabras en pantallas y reportes. Parte de este proceso también se denomina "personalización" o "parametrización", cuando el Software en cuestión esta preparado para modificar su comportamiento en base a variables; a veces parte de la localización se realiza a través de esta parametrización lo cual reduce el esfuerzo de modificación requerido.</p>	

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: El proyecto puede llegar a demorar más de lo planificado en su localización y afectar el plazo de entrega acordado.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Preguntar en el momento de la preventa si el paquete o software importado ya ha sido instalado anteriormente en el país. Si no es el caso asegurarse de que se le asigne un tiempo importante a las tareas de localización.</li> </ul>	

**Enlatados: ajustar mi proceso en vez del software**

DESCRIPCIÓN	
<p>Cuando se adquiere un Software "enlatado" significa que se debe utilizar sin modificar o a lo sumo ajustándolo a nuestras necesidades a través de parámetros. A veces el cliente ve el funcionamiento del Software enlatado y decide su adquisición pero al momento de implantarlo ya sea él o su personal, pretenden que el software se adapte al proceso del cliente. Esta práctica es muy poco recomendable pues usualmente el Software en cuestión ha sido construido sobre la base de determinado proceso (estándar) y modificarlo implica cambiar su filosofía, lo cual puede llegar a ser más caro que construir el sistema nuevamente.</p>	

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: El proyecto indefectiblemente no será entregado en el plazo acordado.</li> <li>○ Costo: el proyecto tendrá un costo muy superior al presupuestado.</li> <li>○ Alcance: Al cambiar el proceso implícito en el Software es probable que se pierdan ciertas funcionalidades.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Decide durante la preventa si está dispuesto a modificar el proceso de su empresa en línea con el software enlatado. De no ser así, desista de adquirir el software.</li> </ul>	

**Las modas de management y de la tecnología**

DESCRIPCIÓN	
<p>A través de conferencias de los Gurús del Management o publicaciones especializadas, los directivos de las empresas se ven bombardeados de información sobre nuevas teorías o modas. Alguna de estas modas generan verdaderas estampidas en los ejecutivos que se entusiasman e intentar implementar rápidamente la nueva corriente de pensamiento. A veces estas modas impactan al mundo de la tecnología. Por citar un caso esta el Downsizing, que originó una importante migración de los sistemas Mainframe (alta gama) a los sistemas Medios (conocidos como Unix). A algunas empresas el cambio le fue positivo y algunas sufrieron pérdidas; obviamente después surgió la corriente del Rightsizing o sea que cada uno tiene que tener sistemas del porte adecuado. Un proyecto informático puede tener su origen en una de estas modas. Otro ejemplo es el del Open software: es razonable considerar el adoptar un sistema operativo como Linux y gradualmente ir incorporando Software Libre, pero muy distinto es intentar liderar la corriente y pretender reemplazar de un día para el otro al software propietario por todo software libre.</p>	

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance: Puede ser que el cliente no alcance a obtener los resultados que busca con el proyecto.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Dejar que otros experimenten con las modas. Verificar los resultados reales de otras compañías con proyectos originados con alguna corriente de pensamiento nueva.</li> </ul>	

Sistemas paramétricos y paradigmas distintos	
<p>Es común pensar que cuando uno adquiere un sistema paramétrico, tendrá parámetros para todo, pero en realidad esto no es siempre así, especialmente si el sistema procede de otra cultura con otros paradigmas. Tomemos un caso real para exemplificar esto: considere un Gerente de sistemas de un Banco Latinoamericano, cuando le ofrecen un sistema paramétrico él asume que entre los miles de parámetros habrá uno que determine a partir de qué monto un cajero de una sucursal le solicita autorización a su supervisor para pagar un cheque. Cuando le ofrecen un sistema paramétrico que proviene de Dinamarca, él ignora que en Dinamarca pagan un cheque sin pedir autorización, si la firma es correcta y hay saldos disponibles pagan, no importa si se trata de un millón de dólares. Obviamente el sistema de Dinamarca no tiene un parámetro y menos la funcionalidad de autorizaciones de supervisores en línea.</p>	

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance: Según el grado de diferencia entre los paradigmas, puede no estar toda la funcionalidad esperada o se puede llegar a cancelar el proyecto.</li> </ul>	

RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ En el momento de preventa investigue si el Software ha sido instalado en su país anteriormente.</li> <li>○ Haga todas las preguntas obvias sobre funcionalidad antes. Invite a participar a sus empleados más escépticos con los sistemas, son los que más preguntas obvias tendrán.</li> </ul>	

**Arquitectura de la solución**

DESCRIPCIÓN	
El tema de arquitectura es muy extenso pero intentaremos sintetizarlo. Frecuentemente a los clientes no les interesa mucho la arquitectura de una solución y en las actividades de preventa, se muestran unos diagramas pero se pone más énfasis y atención en las pantallas que se muestran a modo de ejemplo. Pero las arquitecturas no dan lo mismo, por ejemplo una solución puede ser cliente servidor o ser una solución WEB, esto trae aparejado los lenguajes que se utilizarán, el equipamiento necesario y habrá aspectos de confiabilidad y seguridad que variarán.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: El proyecto entregado puede no satisfacer las expectativas del cliente.</li> <li>○ Costos: Se puede llegar a pagar más de lo requerido para la alternativa elegida.</li> </ul>	<ul style="list-style-type: none"> <li>○ Es necesario que el cliente cuente con asesoramiento técnico para disminuir el déficit de conocimiento tecnológico.</li> <li>○ Se puede pedir a varios proveedores propuestas tecnológicas para resolver las necesidades del cliente, dejando que cada uno proponga la arquitectura técnica de su preferencia.</li> </ul>

**Fanáticos de una tecnología**

DESCRIPCIÓN	
Algunos proveedores en sus acciones de preventa o algunos responsables de tecnología de algunos clientes suelen tener una marcada inclinación hacia determinada metodología. Esto es inconveniente pues los primeros tienden a enfatizar demasiado las bondades de la tecnología de su predilección sin mencionar los puntos débiles. Los segundos intentarán demoler a las propuestas que no correspondan con la metodología de su predilección, aumentando el impacto de las debilidades de las otras tecnologías.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: El proyecto no satisface las expectativas.</li> </ul>	<ul style="list-style-type: none"> <li>○ Para un caso se recomienda el recibir distintas propuestas de varios proveedores a fin de identificar distintas tecnologías y contraponer los argumentos de un proveedor con otro.</li> <li>○ Para el caso de una predilección del responsable de sistemas es más difícil de identificar y evitar. Si se identifica esta situación se puede solicitar la opinión de un consultor externo.</li> </ul>

**Desventajas de un proveedor de demasiado peso**

DESCRIPCIÓN	
Nos referimos a un proveedor de demasiado peso cuando se trata de alguna de las cinco mayores firmas de consultoría en el mundo o los fabricantes de Hardware y Software de reconocimiento mundial a través de los años. Cuando realizamos un proyecto con estas firmas nuestra confianza aumenta y nuestros cuestionamientos técnicos disminuyen. Estas firmas tienen procesos de calidad implantados pero eso no quita que puedan cometer errores.	
EFFECTOS	RECOMENDACIÓN

**○ Plazo, Costo o Alcance según la situación.****RECOMENDACIÓN**

- Cuestionar todo durante la preventa.
- De ser posible que ellos integren la solución y/o se firme un contrato Llave en mano.

**Tecnologías robustas y confiables versus tecnologías ágiles**

DESCRIPCIÓN	
Existen tecnologías diversas que se pueden emplear en un proyecto informático. Se debe tener en claro que se debe balancear las necesidades del cliente con las posibilidades de la tecnología. Es conocido que los sistemas operativos de las computadoras personales a veces tienen un fallo y se detienen. Entonces la pregunta es si confiarían una necesidad que no puede detenerse (imaginemos riesgo de vida) a un sistema de este tipo.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: elegir una tecnología no adecuada puede significar insatisfacción del cliente por fallas en la continuidad del servicio.</li> </ul>	<ul style="list-style-type: none"> <li>○ Hacer un buen balance entre los riesgos de cada tecnología y las necesidades del cliente</li> </ul>

**Mantenibilidad de la aplicación**

DESCRIPCIÓN	
Cuando se desarrolla una solución a medida, no es suficiente con que la aplicación funcione correctamente al ser implementada. Adicionalmente se debe considerar cómo está construida la aplicación en el sentido de cuán caro será realizarle modificaciones al código ya sea para introducir mejoras o para arreglar errores. ¿Está documentado el nuevo sistema?, ¿existe una descripción de qué es lo que hace cada componente?, ¿los programas tienen una estructura que permita entenderlos por un tercero?, ¿las variables tienen nombres que los relacionan con su contenido?, ¿el lenguaje de programación es conocido y existen muchos programadores disponibles?, etc.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Costo: el costo de los mantenimientos posteriores podría resultar muy costoso.</li> </ul>	<ul style="list-style-type: none"> <li>○ Asegurarse que el lenguaje de programación sea universalmente conocido y que el proveedor utilice una metodología orientada a la calidad.</li> </ul>

**Limpieza de Datos**

DESCRIPCIÓN	
En ocasiones el proyecto informático reemplazará sistemas existentes los cuales tienen una cantidad de información que se tiene que convertir al nuevo sistema. Puede ser crítico contar con estos datos y puede ser que las bases de datos sean muy voluminosas. Generalmente durante las pruebas, cuando se hayan empleado los programas de conversión de datos, se descubra que la calidad de los datos de los sistemas antiguos hace que los programas cancelen o que muchos registros sean rechazados por los programas de conversión. El proveedor le indicará al cliente que él necesita mejorar sus datos y el cliente descubrirá que el proceso manual le puede demorar un año y que contratar servicios de limpieza de datos (File Clearing & File Cleansing) que utilizan herramientas especializadas, puede ser muy oneroso y también implica plazos considerables.	
EFFECTOS	RECOMENDACIÓN

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: El proyecto seguramente no terminará en el plazo previsto.</li> <li>○ Costo: Los costos de depuración de datos serán elevados.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Asegurarse que el proveedor en el periodo de preventa acceda a los datos y realice muestreos a fin de determinar si será necesario realizar una limpieza de los datos históricos.</li> </ul>	

**No tener "lecciones aprendidas"**

DESCRIPCIÓN	
<p>En los proyectos informáticos, especialmente en aquellos en los que se construye una solución a medida, los proveedores de mayor prestigio basan su accionar en dos pilares: a) una metodología sólida; b) buenos profesionales. En un mercado creciente, los proveedores están llenos de nuevos profesionales; si su metodología solo se limita a describir cuál es el proceso a seguir, es posible que no tengan registro de los errores cometidos en otros proyectos, por lo tanto si nos tocan profesionales nuevos o sea sin experiencia (donde experiencia es igual a lecciones aprendidas, o sea cúmulo de errores que no se deben repetir), es muy probable que se repitan situaciones que en este documento se detallan.</p>	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance, Plazo y/o Costo de acuerdo a las circunstancias.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ En el periodo de preventa preguntarle al proveedor cómo evitan volver a cometer errores de otros proyectos. Si la respuesta es mediante la metodología, pregúntenle si la metodología incluye lecciones aprendidas, o sea si un proyecto encontró que el sistema operativo XXX no permitía al sistema de base de datos YYY hacer recuperación de datos, cómo se entera quien está en el siguiente proyecto con XXX e YYY que este error ocurrió.</li> <li>○ Si las respuestas no son satisfactorias, se sugiere solicitar proyectos llave en mano.</li> </ul>	

**Construir o comprar**

DESCRIPCIÓN	
<p>Este es uno de los eternos dilemas de las áreas de sistemas. Construir es más interesante y permite desarrollar funciones justo a la medida del usuario pero el costo (a veces) y plazo pueden ser excesivos. Comprar es más rápido y seguro pero puede existir una brecha entre los requerimientos originales y lo que se provee finalmente. A veces también el costo de comprar es demasiado elevado, depende de qué software estemos considerando.</p>	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Costo, Plazo o Alcance dependiendo de la alternativa que se elija.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Primero es importante tener presente que existen las dos alternativas.</li> <li>○ Segundo hacer un análisis profundo de los pro y contras de cada alternativa.</li> </ul>	

**No contar con recuperación ante desastres (Disaster Recovery Planning)**

DESCRIPCIÓN	
-------------	--

Si un proyecto informático brinda una solución a un área crítica del cliente y esta genera una total dependencia sobre el nuevo sistema, es importante que se consideren medidas de recuperación ante desastres, pues en caso de ocurrir un desastre y más allá del costo del mismo, las consecuencias económicas que generaría el área crítica podrían significar la quiebra de la compañía.

Después del 11 de septiembre, muchas firmas que tenían sus oficinas centrales en las torres gemelas establecieron sus operaciones en Nueva Jersey, demorando en muchos casos entre 48 y 96 horas solamente.

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Costos, según se explicó antes.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Contratar una consultoría sobre recuperación de desastres que se encargará de</li> <li>○ Identificar los registros vitales (archivos a partir de los cuales se pueden reanudar operaciones, lista de clientes y teléfonos, pedidos pendientes, precios de todos los productos de nuestros proveedores, etc.).</li> <li>○ Establecer un procedimiento para almacenar los registros vitales en forma periódica en un lugar con distintos riesgos que los del lugar donde se realizan las operaciones.</li> <li>○ Identificar y entrenar a las personas clave de la organización (y sus reemplazos) que deberán actuar ante un desastre.</li> <li>○ Tener un plan de acción ante desastres.</li> <li>○ Hacer simulacros una vez al año.</li> <li>○ Identificar un lugar sustituto donde realizar las operaciones.</li> <li>○ Contratar servicios de procesamiento de datos contingente.</li> </ul>	

**No contar con Infraestructura de comunicaciones**

DESCRIPCIÓN	
<p>Este punto es similar al de <i>capacity planning</i>. Se puede implementar un proyecto que tiene una dependencia en la infraestructura de comunicaciones, ya sea que esta sea la del cliente o la del país. Si no se satisface dicha dependencia, el proyecto no será de utilidad.</p>	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance: No se logran los resultados esperados por el proyecto pues la infraestructura no da abasto.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ En el momento de la preventa se debe identificar esta dependencia, se debe estimar la demanda inicial y el crecimiento de la demanda y se deben evaluar las alternativas como para contar con dicha infraestructura y de ser necesario considerar los costos dentro del proyecto.</li> </ul>	

**Proyectos multipaís multihorario**

DESCRIPCIÓN	
Algunos proyectos informáticos grandes se realizan con proveedores o entidades de otros países, tales como Factorías de Software ( <i>Software Factories</i> ). Se debe tener esto en cuenta no solo por el tema del idioma (en caso de que usen más de uno) sino de las diferencias horarias, que varían durante el año y que pueden hacer disminuir las ventanas de comunicación considerablemente. Los peores casos se dan al trabajar con países del Pacífico, en donde las teleconferencias pueden iniciarse a las 20 hs de un país latinoamericano lo que equivale a las 8 de la mañana en Sydney. Existen empresas de la India que ya tienen solucionado esto, teniendo un rol de "enlace de comunicaciones" dentro del proyecto cubierto por tres analistas que trabajan 8 horas cada uno cubriendo las 24 horas del día.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: Las limitaciones de horario generalmente generan demoras en los plazos.</li> </ul>	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Tenga presente este factor durante la preventa como para que se tomen los recaudos necesarios.</li> </ul>	

**Terminología común en contrato y proyecto**

DESCRIPCIÓN	
De nada sirve que un contrato sea redactado por un abogado que desconoce la terminología informática pues el contrato resultante no podrá ser utilizado como base del acuerdo durante la ejecución del proyecto, ya que los ingenieros encargados del proyecto no entenderán el alcance de las cláusulas y términos empleados.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance: Es altamente probable que parte de los compromisos especificados en el contrato no se satisfagan si se dan las condiciones antes detalladas.</li> </ul>	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ En la medida que coincida la terminología del contrato y del proyecto, las personas que llevan adelante el proyecto podrán comprender los lineamientos y compromisos incluidos en el contrato.</li> <li>○ Se recomienda que los ingenieros participen en la redacción del contrato a fin de que tengan un total entendimiento de los compromisos que se asumen.</li> <li>○ Se recomienda incorporar en los Anexos Técnicos de los contratos una sección de Glosario y Abreviaturas a fin de permitir el uso de términos técnicos muchas veces en idioma inglés.</li> </ul>	

**Bajo nivel de usuarios**

DESCRIPCIÓN	
Se puede definir un excelente proyecto, tomar todos los recaudos y no llegar a considerar a las personas que terminarán utilizando el sistema. Es común para los profesionales informáticos que les sea difícil darse cuenta que cualquier funcionalidad muy sencilla puede llegar a ser imposible de utilizar por personas desacostumbradas a la utilización de la tecnología. La brecha digital genera sospechas, dudas y miedos en aquellos que no están familiarizados con la tecnología. No es necesario viajar a zonas no urbanas para encontrar estas situaciones, existen oficinistas en los grandes centros urbanos que no saben operar una máquina de fotocopias y son justamente esos oficinistas los futuros usuarios de los sistemas informáticos que se implementan.	

EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance y Tiempo: Los usuarios no podrán hacer uso de todas las bondades del nuevo sistema, seguramente se limitarán a utilizar el sistema de la manera más básica, después de someterse a capacitaciones adicionales que seguramente demorarán la entrega del proyecto.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Conocer antes de planificar un proyecto, cuál es nivel de usuarios que utilizarán el sistema y diseñar la solución y el proyecto acordes con el nivel identificado.</li> <li>○ Solicitar la incorporación de personas con mayor capacidad para integrar el grupo de trabajo del cliente, que lideren la utilización de las nuevas soluciones y permanezcan como referentes.</li> </ul>	

**Mala Comunicación**

DESCRIPCIÓN	
Cada vez que se analizan las deficiencias en una organización o en un proyecto, se identifican fallas asociadas a la comunicación. Gracias a la mala comunicación el equipo de proyecto puede desconocer el contrato, los empleados del cliente pueden desconocer que se realizará un proyecto, los miembros del equipo pueden no tener muy claro el objetivo del proyecto, el plazo comprometido, el estado del proyecto o el plan del proyecto.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance y Tiempo: por mala comunicación no se cumplen las tareas en tiempo y se deja de satisfacer los requerimientos del cliente.</li> </ul>	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Obligar a la conducción del proyecto a que lea y tenga presente el contrato del proyecto.</li> <li>○ Realizar una reunión de lanzamiento con el personal del cliente y el equipo del proyecto en donde se explique el proyecto, su objetivo, los plazos, los roles de cada uno y los beneficios que aportará la solución.</li> <li>○ Realizar reuniones periódicas con los distintos niveles del cliente para mantenerlos informados de la marcha del proyecto.</li> <li>○ Realizar reuniones semanales con el equipo del proyecto para revisar el avance, los problemas, los riesgos y tener una comunicación bidireccional efectiva.</li> <li>○ Forzar a que se realicen y distribuyan las minutos de las reuniones en especial aquellas en que se llegan a acuerdos o se asumen compromisos.</li> </ul>	

**PROBLEMAS TÉCNICOS**

Las siguientes cuestiones se relacionan con aspectos de Ingeniería de Software o con la Metodología de Proyectos Informáticos. Son aspectos técnicos que el proveedor debe tener en cuenta y no está de más que el cliente los conozca para mejorar el criterio con el cual evalúa a su proveedor. Estas falencias se deben identificar y resolver durante la ejecución del proyecto para evitar sus consecuencias perjudiciales para el éxito del proyecto:

1. No tener suficientes (variedad de) pruebas
2. No tener un plan de calidad
3. No tener un grupo de pruebas independiente
4. No contar con clasificación ni cuantificación de errores
5. No tener coordinación entre grupos

6. No tener control presupuestario
7. No tener un Flujo de Fondos (*Cashflow*) del proyecto
8. No hacer reportes de avance
9. No prever problemas de infraestructura
10. No prever problemas de licencias necesarias para desarrollo
11. No tener un plan
12. No actualizar el plan
13. Grandes barras y síndrome del 90%
14. Quién controla lo que hace el programador
15. No identificar prerequisitos
16. No identificar dependencias críticas o entregables críticos
17. No identificar camino crítico
18. Plan sin plazos reales
19. Plan sin horas
20. Roles predefinidos
21. Criterios de estimación inadecuados, incompletos, optimistas o ingenuos
22. No considerar ni manejar los riesgos
23. No tener requerimientos técnicos
24. Tener Requerimientos que no se pueden probar
25. No tener control de cambios
26. No tener trazabilidad (controlar que se diseñen, construyan y prueben todos los requerimientos)
27. Tiempo de respuesta
28. Comenzar a construir sin requerimientos claros
29. No tener control de configuración
30. No tener documentación
31. No tener control de documentación
32. No controlar a los proveedores
33. No realizar pliegos técnicos con los proveedores
34. Conversión de nombres
35. Estándares de programación
36. Técnicas de análisis, diseño y pruebas
37. Falta de modelo de datos
38. Comenzar sin acuerdo formal
39. No comenzar una etapa sin terminar la anterior

## DESCRIPCIÓN DE LOS PROBLEMAS TÉCNICOS

### No tener suficientes (variedad de) pruebas

DESCRIPCIÓN
Es común que exista una tendencia a concentrar los esfuerzos de los proyectos en la construcción de la solución y no concentrarse mucho en las pruebas. Para colmo si se producen demoras en el proyecto lo más común será reducir el tiempo de pruebas. Las pruebas son un buen método para verificar que el sistema funciona, que cumple con los requerimientos del proyecto y que no tiene errores en la ejecución. Existen varios tipos de pruebas <sup>9</sup> (pruebas Unitarias, de Integración, de Sistema, de aceptación Usuario, de Performance, de Stress, etc.); en la medida que podamos realizar todas las pruebas necesarias brindándoles un tiempo adecuado, podremos eliminar errores y asegurarnos de estar entregando un sistema a satisfacción del Cliente.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Alcance: no realizar todas las pruebas afecta la calidad y por lo tanto el alcance.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Darle mayor importancia a las diversas pruebas que se deben realizar.</li> </ul>

### No tener un plan de calidad

DESCRIPCIÓN
Algunas metodologías de Proyecto recomiendan que exista un plan de calidad. Este consiste en la definición de la estrategia de pruebas, el detalle de las pruebas que se realizarán y todas aquellas acciones a realizar durante el proyecto con el objetivo de prevenir defectos y contribuir a la calidad de la solución.
EFEKTOS
<ul style="list-style-type: none"> <li>○ Alcance: no tener un plan de calidad impacta la misma y por lo tanto el alcance.</li> </ul>
RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Establecer al principio del proyecto la estrategia con respecto a la calidad y las pruebas, estableciendo metas y luego realizando el seguimiento correspondiente.</li> <li>○ No solo realizar pruebas sino acciones preventivas como           <ol style="list-style-type: none"> <li>1. Revisiones de documentos de requerimientos y de diseño</li> <li>2. Realizar revisiones de pares al proyecto</li> <li>3. Tener grupos independientes (del mismo proveedor) que revisen (auditen) al proyecto</li> </ol> </li> </ul>

### No tener un grupo de Pruebas independiente

DESCRIPCIÓN
Es fácilmente comprobable que las mejores pruebas las realizan personas distintas a aquellas que construyeron determinado sistema. Cuando alguien construye algo interpreta requerimientos, si hubo un error en dicha interpretación se repetirá cuando la misma persona pruebe su sistema. Un tercero a quien se le encarga una prueba en general es más creativo, más exhaustivo y malicioso intentando "romper" el sistema. Su objetivo es encontrar un error y por lo tanto se debe entender que un caso de prueba exitoso es aquel que logra identificar un error y no al revés. Cuando el mismo autor prueba su programa está intentando demostrar lo bien que lo construyó y por lo tanto es muy probable que no sea tan exhaustivo.

<sup>9</sup> Se incluye un anexo en el cual se describen entre otros temas, las diversas pruebas que se deben realizar.

EFFECTOS	
○ Alcance: No contar con un grupo independiente de pruebas afecta la calidad y por lo tanto el alcance.	
RECOMENDACIÓN	
○ Contar con un grupo de pruebas independiente que asegure una mejor prueba. Si no se dispone de un grupo independiente y se ha construido la solución con más de un equipo, se pueden intercambiar los equipos para que cada uno pruebe lo que el otro equipo construyó.	

**No tener coordinación entre grupos**

DESCRIPCIÓN	
Esta cuestión es muy básica y obvia pero lamentablemente demasiado frecuente. Por ejemplo en un proyecto con varios equipos, un equipo incorpora un cambio y no le avisa al otro equipo; un equipo está seguro que se va a atrasar y no le avisa al equipo que espera recibir un componente en fecha.	
EFFECTOS	
○ Alcance y Tiempo: cuando existe coordinación entre equipos de trabajo es muy difícil no impactar el alcance del proyecto y poder terminarlo en los plazos comprometidos.	
RECOMENDACIÓN	
○ Asegurarse que existan instancias formales que fomenten la comunicación y coordinación de los equipos.	

**No tener control presupuestario**

DESCRIPCIÓN	
Dependiendo de cómo se haya realizado el contrato, la falta de control presupuestario afectará al proveedor o al Cliente. Si se trata del proveedor, al cliente también le interesa, pues a ningún cliente le interesa que sus proveedores sufran perjuicios económicos que quizás los hagan ir a la quiebra.	
EFFECTOS	
○ Costo: La falta de control presupuestario tendrá un impacto directo sobre el costo del proyecto.	
RECOMENDACIÓN	
○ En el momento de planificar el proyecto, estimar los costos mensuales que se deben afrontar. Al ejecutar el Proyecto se debe realizar un seguimiento económico controlando los costos reales y comparándolos con los planificados.	

**No tener un Flujo de Fondos (Cashflow) del Proyecto**

DESCRIPCIÓN	
Tener un flujo de fondos proyectado es necesario para determinar los momentos en los cuales se deberá contar con fondos adicionales para financiar el proyecto. Esta práctica es recomendable tanto para el cliente como para el proveedor. Nuevamente al Cliente le conviene que el proveedor tenga un flujo de fondos proyectado, pues de no hacerlo es posible que durante la ejecución del proyecto se quede sin suficientes fondos y por ejemplo falle en pagarle el sueldo a sus empleados, los mismos que trabajan en el proyecto.	
EFFECTOS	

○ Alcance: Problemas económicos pueden ocasionar que se cancele el proyecto anticipadamente.	
RECOMENDACIÓN	
○ Preparar en los momentos de planificación del proyecto, el flujo de fondos correspondiente, utilizando tiempos realistas en relación al proceso de cobro de facturas.	

**No hacer reportes de avance**

DESCRIPCIÓN	
Como se indicó anteriormente, la falta de comunicación puede ser fatal en un proyecto. Los reportes de avance son necesarios para que el cliente esté enterado sobre la marcha del proyecto y conozca si debe tomar acciones sobre la participación de sus empleados en el proyecto. Así mismo el confeccionar un reporte de avance fuerza al equipo de proyecto a tomar conciencia en el caso de que existan demoras, especialmente si estas se van acumulando.	
EFFECTOS	
○ Tiempo: En general la falta de reportes de avance no deja ver demoras y el proyecto termina fuera del plazo comprometido. ○ Alcance: Sin reportes de avance pueden producirse incumplimientos que afecten la disponibilidad de ciertas funciones del sistema en los plazos comprometidos.	
RECOMENDACIÓN	
○ Realizar reportes de avance haciendo referencia al plan del proyecto y aportando medidas numéricas y realistas del avance.	

**No prever problemas de infraestructura**

DESCRIPCIÓN	
Esta cuestión es demasiado amplia, puede fallar parte del entorno técnico (un disco, un resguardo, cortes en las transmisiones, corrupción de índices, etc.), o pueden no servir los elementos que estaban planificados (la base de datos tarda mucho tiempo en recuperar la información, no maneja bien la concurrencia, etc.), o pueden no estar los elementos planificados (se dependía de un equipamiento que demora en llegar al país). El autor ha llegado a conocer un proyecto en el cual el servidor en donde residía la aplicación se cayó, con la aclaración inmediata que físicamente se cayó de un segundo piso a un primer piso y quedó destruido!	
EFFECTOS	
○ Tiempo: la falta de infraestructura necesaria en general demora o detiene el proyecto, no pudiendo entregarse la solución en el plazo comprometido.	
RECOMENDACIÓN	
○ Prever alternativas para enfrentar problemas de infraestructura.	

**No prever problemas de licencias necesarias para el desarrollo**

DESCRIPCIÓN	
Cuando se adquieren licencias para determinado software en realidad se recibe una autorización para utilizar dicho software. Este derecho de uso en general viene condicionado a la utilización en un solo equipo, en especial si se trata de un sistema operativo o una base de datos (ya que otro tipo de software requiere permisos de cierta cantidad de licencias que se instalan en un servidor y son de uso concurrente). Si un proyecto se desarrolla en un equipo (Host o Servidor central) distinto al de producción, será necesario contar con más de una licencia o un acuerdo con el proveedor del software. Este detalle es importante pues hay licencias con un costo de cientos de miles de dólares.	
EFFECTOS	

EFFECTOS	
○ Costo: Licencias adicionales no previstas tendrán un impacto en el costo del proyecto. Dependiendo de cómo está armado el contrato esto afectará al proveedor o al cliente.	
RECOMENDACIÓN	
○ Tener en cuenta la necesidad de licencias adicionales para el desarrollo al momento de planificar el proyecto.	

**No tener un plan**

DESCRIPCIÓN	
En un proyecto no tener un plan es fatal, es la receta para el fracaso. Un plan es necesario para tener un orden para realizar las tareas de un proyecto, cada persona del equipo de proyecto sabe qué hacer, se pueden determinar las secuencias de actividades, aquellas actividades que se pueden hacer en paralelo y aquellas que se pueden realizar en cualquier momento o con determinada holgura de tiempo. Con un plan se sabe quién es el responsable de cada tarea y se sabe en qué fecha se deben alcanzar hitos intermedios. Cuando hay demoras se puede analizar el plan para ver alternativas y/o calcular cuál será la demora final.	
EFFECTOS	
○ Tiempo: El principal impacto de no tener un plan es seguramente no cumplir con el plazo de entrega comprometido.	
RECOMENDACIÓN	
○ Al comenzar el proyecto realice un ejercicio de planificación detallada siguiendo "buenas prácticas" en la construcción del plan.	

**No actualizar el plan**

DESCRIPCIÓN	
Es común que los responsables de un proyecto comiencen con un plan pues es inadmisible pretender realizar un proyecto sin un plan, pero no actualizan el plan cuando al poco tiempo comienzan a surgir cambios al proyecto ya sea por nuevos requerimientos, cambio a requerimientos o porque surgieron demoras y se identifican cambios en la secuencia de actividades a seguir. Un plan que no se actualiza pierde validez rápidamente.	
EFFECTOS	
○ Tiempo: No tener un plan actualizado es equivalente a no tener un plan y es altamente probable que no se cumplan los plazos comprometidos.	
RECOMENDACIÓN	
○ Ante cada cambio en el proyecto es imprescindible actualizar el plan y distribuirlo a los miembros del equipo.	

**Grandes barras y síndrome del 90%**

DESCRIPCIÓN	
Existen profesionales que cumplen con generar un plan y mantenerlo actualizado, pero las actividades aparecen cubriendo grandes períodos de tiempo, que al graficar el plan aparecen como grandes barras de tiempo. Esta práctica es muy nociva pues por lo general los informes de avance que se darán sobre una actividad planificada de esta manera, no dejarán en claro si dicha actividad está realmente avanzando de acuerdo al plan o no. Por lo general se reporta un avance normal para dicha actividad y al llegar al final aparece primero que están al 90%, el próximo informe de avance estarán al 95%, el próximo al 97% y así se irán estirando los plazos sin que la actividad esté realmente finalizada.	

EFFECTOS	
○ Tiempo: Con esta práctica se encubren las demoras de actividades intermedias que terminan impactando la fecha final comprometida	
RECOMENDACIÓN	
○ Planificar con tareas de mínimo 8 horas y máximo 40 horas o en forma equivalente, que cada tarea no duré más de una semana. De esta manera si se pretende encubrir una demora esto solo será posible hacerlo durante una semana.	

**¿Quién controla lo que hace el programador?**

DESCRIPCIÓN	
En los grandes proyectos informáticos se pone mucho énfasis en la gestión del proyecto y/o se puede estar subcontratando reiteradamente (El Proveedor principal subcontrata al proveedor A quien subcontrata ciertas tareas al proveedor B...) poniendo mucha distancia entre la conducción del proyecto y quienes construyen la solución, o sea los programadores. Es común entonces que para un reporte de avance se le pregunte al programador cómo está con el trabajo y este responda que terminó o "casi listo", pero nadie revisa si lo que el programador terminó está completo, si las pruebas del programador fueron efectivas, si el programador utilizó los estándares oficiales, etc.	
EFFECTOS	
○ Alcance y Tiempo: por lo general la falta de control sobre los programadores impacta la Calidad y termina produciendo demoras.	
RECOMENDACIÓN	
○ No solo poner foco en la gestión general del proyecto sino en los niveles intermedios de control. ○ Evitar tener subcontrataciones de subcontrataciones. ○ Asegurarse que los supervisores de los programadores sigan un proceso estándar de revisión de sus colaboradores.	

**No identificar prerequisitos**

DESCRIPCIÓN	
Si no se identifican prerequisitos de determinadas etapas o tareas, en plena ejecución el equipo se encontrará que no tiene posibilidad de avanzar con el proyecto, si dichos prerequisitos no están disponibles.	
EFFECTOS	
○ Tiempo: Una demora de un prerequisito se transforma en demoras sobre el proyecto que pueden impactar la fecha de entrega comprometida	
RECOMENDACIÓN	
○ Identificar todos los prerequisitos, ya sea de etapas o tareas, inclusive en caso de que sean obvios y hacerles seguimiento para asegurar su disponibilidad en las fechas comprometidas.	

**No identificar dependencias críticas o entregables críticos**

DESCRIPCIÓN	
En un gran proyecto informático se generan cantidad de dependencias críticas entre los distintos grupos del proyecto. Si no se identifican estas dependencias, no se les puede hacer seguimiento, no hay conocimiento y conciencia de los grupos que lo que están desarrollando es crítico para otro equipo en determinada fecha. Si se sabe que hay demoras en un entregable crítico, con bastante anticipación se podrán evaluar alternativas y replanificación de actividades.	

EFFECTOS	
<ul style="list-style-type: none"> <li>• Tiempo: No identificar dependencias críticas en general termina generando demoras que pueden impactar el plazo comprometido.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>• Identificar las dependencias críticas en cada plan. Si existen planes separados para distintos grupos de proyecto, dichos planes deberán tener ambos las dependencias críticas marcadas en forma común (coincidiendo fecha de entrega).</li> </ul>	

**No identificar el camino crítico**

DESCRIPCIÓN	
<p>El camino crítico en un plan está dado por la secuencia de tareas que no tienen holgura de tiempo para su ejecución. Dicho en otras palabras, si una de las tareas del camino crítico sufre una demora, el proyecto no finalizará en el plazo comprometido. Si tenemos un plan pero no identificamos y controlamos el camino crítico, no vamos a tener conciencia cuando se produzca una demora que el proyecto se atrasará indefectiblemente.</p>	
EFFECTOS	
<ul style="list-style-type: none"> <li>• Tiempo: como se argumentó en el párrafo anterior, no se cumplirá con el plazo comprometido</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>• Identificar el camino crítico de un proyecto y controlar las tareas del camino crítico durante la ejecución del proyecto, protegiéndolas en lo posible para evitar que se demoren.</li> </ul>	

**Plan sin plazos reales**

DESCRIPCIÓN	
<p>Si el proyecto tiene un plan es muy útil ir incorporándole las fechas de inicio y término reales, de esta manera el proveedor podrá ver y mostrar en los avances del proyecto cuál es la ejecución del mismo versus la planificación original. Esto es conveniente para ver por ejemplo que todas las tareas que finalizaron lo hicieron 2 ó 3 semanas pasada la fecha original, lo cual marca una tendencia que se debe tener en cuenta para tomar acciones correctivas o acordar una nueva fecha de finalización. No hacer esto genera un falso efecto de que todas las tareas terminadas fueron ejecutadas dentro de los plazos originales.</p>	
EFFECTOS	
<ul style="list-style-type: none"> <li>• Tiempo: de existir demoras continuas estas serán tenidas en cuenta y es muy probable que no se cumpla con el plazo comprometido.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>• Incorporar las fechas reales de inicio y de fin al plan durante la ejecución del mismo.</li> </ul>	

**Plan sin horas**

DESCRIPCIÓN	
<p>En un plan es importante identificar el costo de cada tarea, generalmente expresado en horas de esfuerzo, tanto en el plan original como en la ejecución de cada tarea. No hacerlo puede significar que el proveedor no se ha tomado el tiempo como para estimar el esfuerzo de cada tarea y durante la ejecución del proyecto se puede descubrir que los recursos del proyecto no alcanzan para ejecutar determinadas tareas en el tiempo que se le ha asignado.</p>	

EFFECTOS	
<ul style="list-style-type: none"> <li>• Tiempo: Si no alcanzan los recursos, las tareas demorarán más de lo planificado y por lo tanto no se cumplirá con la fecha comprometida.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>• Incluir la cantidad de horas tanto en el plan como en la ejecución de cada tarea.</li> <li>• Construir un plan de recursos (personígrafo o Staffing plan) que confirme que la cantidad y perfil de personas dedicadas al proyecto en cada momento serán suficientes para cubrir las horas requeridas en el plan.</li> </ul>	

**Roles predefinidos**

DESCRIPCIÓN	
<p>En todo proyecto es necesario que los roles estén bien definidos y que todos los roles estén asignados. Un proyecto de magnitud como suele ser un proyecto de Integración de Sistemas involucra un grupo frecuentemente numeroso de personas. Estas personas tienen diferentes profesiones y habilidades y, además, pertenecen a diferentes organizaciones (Integrador, Subcontratistas y Cliente). Por lo expuesto, se trata de un grupo frecuentemente heterogéneo, con culturas y modos de trabajo diferentes. Por ese motivo, es necesario que todos sus integrantes tengan una clara idea de cuál es su misión, sus objetivos y sus responsabilidades en el proyecto. Para cumplir con este concepto es necesario establecer una organización de manejo clara, sencilla, eficiente y que sea conocida y comprendida por todos los integrantes del grupo de trabajo, el cliente y los subcontratistas. Si bien la organización puede diseñarse desde diferentes puntos de vista, una parte de dicha organización es normalmente fija para cualquier proyecto. Esta es la compuesta por el Gte. de Proyecto, el Gte. de Desarrollo, el Gte. de Pruebas, el Arquitecto del proyecto:</p>	
<ul style="list-style-type: none"> <li>• El Gerente de Proyecto es el único responsable ante el cliente y el integrador que el proyecto se complete en tiempo, dentro del presupuesto y con un alto grado de satisfacción por parte del cliente.</li> </ul>	
<ul style="list-style-type: none"> <li>• El Gerente de Desarrollo es el responsable de fabricar todos los componentes de Software del Proyecto.</li> </ul>	
<ul style="list-style-type: none"> <li>• El Gerente de Pruebas es responsable de asegurar que todos los productos fabricados por el proyecto cumplen todos los requerimientos oportunamente acordados con el cliente.</li> </ul>	
<ul style="list-style-type: none"> <li>• El arquitecto del proyecto es responsable por la integridad técnica de los productos fabricados en el proyecto.</li> </ul>	
<ul style="list-style-type: none"> <li>• Estos puestos son ocupados por personal del Integrador, sin embargo, niveles inferiores de la estructura pueden ser ocupados por personal de los subcontratistas o del cliente. De aquí en más, la organización se diseña para cada proyecto específico, pudiendo existir diferentes niveles de acuerdo a la magnitud del proyecto. Por ejemplo, la gerencia de desarrollo puede organizarse por aplicación a instalar (un gerente por cada aplicación), por etapa (un gerente por cada etapa del proyecto, etc.). Puede aplicarse el mismo diseño para la gerencia de pruebas.</li> </ul>	
<p>El arquitecto puede ser una sola persona o un grupo, dependiendo de la magnitud del proyecto. Pueden existir gerencias adicionales a las planteadas: Centro de Cómputos, Telecomunicaciones, Educación, Oficina de Control de Proyecto, etc.</p>	
<p>En todos los casos, deben definirse roles precisos para cada uno de los puestos de la estructura, cualquiera sea ésta. Estas definiciones se registran en el Documento de Roles del Proyecto. Cualquier modificación en la organización (ya sea de estructura o de nombres) debe ser comunicada a toda la organización del proyecto, a los subcontratistas y al cliente.</p>	

<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Tiempo: no tener los roles asignados genera malos entendidos, demoras, afecta la comunicación y en general los plazos del proyecto son los que se ven afectados</li> </ul>	
<b>RECOMENDACIÓN</b>	
<ul style="list-style-type: none"> <li>○ Establecer una estructura de organización de Proyecto.</li> <li>○ Debe estar disponible un Directorio con los nombres y datos de las personas y sus respectivos roles.</li> <li>○ Definir en forma escrita todas las responsabilidades de cada uno de los roles.</li> </ul>	

**Criterios de estimación inadecuados, incompletos, optimistas o ingenuos**

<b>DESCRIPCIÓN</b>	
Un aspecto clave de todo proyecto informático reside en las estimaciones de tiempo y esfuerzo que se realizan en especial en lo relacionado al desarrollo de software.	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Tiempo: Malas estimaciones se transformarán en demoras y en un incumplimiento de los plazos comprometidos.</li> </ul>	
<b>RECOMENDACIÓN</b>	
<ul style="list-style-type: none"> <li>○ Asegurar que se utilicen métodos reconocidos para realizar estimaciones y/o que dichas estimaciones sean revisadas por varias personas.</li> </ul>	

**No considerar ni manejar los riesgos**

<b>DESCRIPCIÓN</b>	
Una práctica muy recomendable en los proyectos Informáticos es tener en cuenta los riesgos y anticipar las alternativas que uno tendría para enfrentar los riesgos. No hacerlo le resta al proyecto la habilidad de maniobrar ante los distintos problemas que surjan y que hayan sido previsibles.	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Tiempo: analizar cada problema y encontrar la mejor solución después que el problema ocurra generará demoras que impactarán la posibilidad de cumplir con los plazos comprometidos.</li> </ul>	
<b>RECOMENDACIÓN</b>	
Durante la preparación de proyectos Informáticos se debe realizar una Evaluación de Riesgos. La evaluación se realiza con la ayuda de una Guía de Evaluación de Riesgos, que lista los hechos o condiciones riesgosas que pueden presentarse en un proyecto de Integración de Sistemas. Existen diversas listas que abarcan situaciones que se deben evaluar, tales como:	
<ul style="list-style-type: none"> <li>• Desarrollo de Hardware ad-hoc para el proyecto</li> <li>• Nivel de definición de requerimientos</li> <li>• Experiencia del Grupo en proyectos similares</li> <li>• Experiencia del Gerente de Proyecto en proyectos similares</li> <li>• Conexiones con sistemas de diversos proveedores</li> <li>• Requerimientos de Performance</li> <li>• Nivel esperado de cambios</li> <li>• Rigidez y/o criticidad del Cronograma</li> <li>• Involucración y dependencias del cliente en el Proyecto</li> <li>• Involucración y dependencias de los subcontratistas en el Proyecto</li> </ul>	

- Términos y condiciones del contrato
- Criterios de Aceptación
- Nivel de planeamiento del Proyecto

Una vez identificados los riesgos, para aquellas situaciones que implican un riesgo medio o alto, se desarrolla un *Plan de Contención*. En este plan se detallan las acciones a tomar (que pueden implicar incluir en el presupuesto del proyecto costos adicionales, incluir en el contrato cláusulas especiales, redefinir criterios de entrada o aceptación, etc.) para mantener el riesgo en un nivel bajo o, si es posible, eliminarlo.

Durante la ejecución del proyecto, periódicamente se revisa la Evaluación de Riesgos, ya que por el propio desarrollo del proyecto pueden aparecer situaciones que impliquen riesgos no detectados oportunamente o, por el contrario, algunos riesgos pueden haber desaparecido.

**No tener requerimientos técnicos**

<b>DESCRIPCIÓN</b>	
Durante el relevamiento de requerimientos en un proyecto informático el proveedor puede omitir preguntarle al cliente por aspectos técnicos y el cliente por falta de conocimientos puede omitir especificar sus necesidades que tengan consecuencias técnicas. Ejemplos de requerimientos técnicos son: tiempo de respuesta, tiempo de procesamiento, volúmenes máximos a manejar.	

<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance: El producto desarrollado puede no satisfacer las expectativas del cliente.</li> </ul>	
<b>RECOMENDACIÓN</b>	
<ul style="list-style-type: none"> <li>○ Interrogar al cliente durante el relevamiento de requerimientos sobre aspectos técnicos a fin de asegurarse cuáles son las expectativas que tiene el cliente sobre la performance técnica de la solución a proveer.</li> </ul>	

**Tener requerimientos que no se pueden probar**

<b>DESCRIPCIÓN</b>	
Una regla de oro es solo comprometerse a satisfacer requerimientos en los proyectos informáticos que puedan ser probados. Caso contrario será muy difícil comprobar que se ha satisfecho el requerimiento cuando la aceptación depende de la opinión de distintas personas y no cuestiones precisas. El ejemplo típico de esta situación es cuando se pide que el sistema sea amigable.	

<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance: Será muy difícil satisfacer requerimientos sobre los que no se pueda llegar a determinar si están cumplidos o no.</li> </ul>	
<b>RECOMENDACIÓN</b>	
<ul style="list-style-type: none"> <li>○ Cuando se presentan este tipo de requerimientos se debe hacer un esfuerzo para traducir los requerimientos en elementos comprobables. En el caso del ejemplo se puede traducir amigable en requerimientos para que el sistema dé mensajes error sin usar códigos, siglas o nombres técnicos, o que la cantidad de información en cada pantalla se mantenga en una determinada cantidad.</li> </ul>	

**No tener control de cambios**

DESCRIPCIÓN	
En un proyecto informático de envergadura, si no se cuenta con un control de cambios lo que sucederá será que distintas personas del cliente irán solicitando e introduciendo en el proyecto cambios a los requerimientos, estos cambios no serán evaluados apropiadamente, no serán comunicados a todo el proyecto para entender si hay otras áreas impactadas, es posible que su incorporación distorsione la arquitectura de la solución, generalmente implicará un mayor costo que no será trasladado al cliente, no se adecuará el plan para reflejar el trabajo adicional y la conducción del proyecto y los ejecutivos del cliente posiblemente no tengan conocimiento que los cambios se están introduciendo.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance y Tiempo: por lo descrito en el párrafo anterior, la solución final posiblemente sea distinta a la que esperan los responsables del cliente y los plazos comprometidos no se cumplen.</li> </ul>	<ul style="list-style-type: none"> <li>○ Establecer un proceso estricto de control de cambios para canalizar todos los requerimientos del cliente que surjan después de haber acordado el alcance del proyecto, evaluar las implicancias de cada cambio en términos de costo y tiempo, poner a consideración del Cliente y/o comité de dirección los cambios solicitados y sus implicancias, ajustar el plan del proyecto y todos los documentos técnicos impactados de acuerdo a los cambios que sean aceptados.</li> </ul>

**No tener trazabilidad (controlar que se diseñen, construyan y prueben todos los requerimientos)**

DESCRIPCIÓN	
Habiendo hecho un esfuerzo al principio del proyecto por tener todos los requerimientos documentados, aprobados por el cliente y con un buen control de cambios para tener esta base siempre actualizada y vigente, es común que no exista un control que confirme que después de construir la solución, todos los requerimientos hayan sido contemplados. Hoy en día, es común que las firmas de auditoría que revisan el trabajo de proveedores de Subsidiarias de compañías extranjeras, exijan un control reiterado sobre la trazabilidad de los requerimientos siendo necesario poder demostrar con qué casos de prueba se verificó cada requerimiento y qué módulos satisfacen cuáles requerimientos. No tener esta práctica permite que en los grandes proyectos informáticos queden requerimientos sin ser satisfechos.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: La funcionalidad final no coincidirá con la totalidad de los requerimientos.</li> </ul>	<ul style="list-style-type: none"> <li>○ Implementar controles en el diseño, construcción y pruebas para confirmar que todos los requerimientos hayan sido atendidos.</li> <li>○ Utilizar números o códigos para identificar a cada requerimiento, cada caso de prueba y armar tablas de referencias cruzadas o incluir las referencias respectivas en todos los documentos.</li> </ul>

**Tiempo de respuesta**

DESCRIPCIÓN	
-------------	--

Aunque ya se han tratado aspectos de requerimientos técnicos se desea poner énfasis en el Tiempo de Respuesta de la solución que genere el proyecto informático, pues este aspecto está siempre visible al usuario y es uno de los factores más comunes de disconformidad. En un proyecto se puede ignorar este aspecto totalmente o se puede tener conciencia del mismo pero asumir que será aceptable o se pueden realizar pruebas pero sin proyectar los volúmenes reales de operación. Todas estas instancias son insuficientes y es posible que la solución que se genere tenga, el día de entrega o después de unos meses, un Tiempo de Respuesta inaceptable.

EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: Un sistema con mal tiempo de respuesta es concebido como de baja calidad y por lo tanto es el alcance el aspecto afectado.</li> </ul>	<ul style="list-style-type: none"> <li>○ Conocer las expectativas del cliente (directivos y usuarios finales) con respecto al tiempo de respuesta que consideran apropiado para la solución que proveerá el proyecto informático y determinar los valores límite a ser utilizados en las pruebas.</li> <li>○ Realizar pruebas con bases de datos pobladas con volúmenes reales de operación y comprobar el tiempo de respuesta.</li> <li>○ Realizar pruebas de Stress del sistema midiendo tiempos de respuesta.</li> </ul>

**Requerimientos: Comenzar a construir sin requerimientos claros**

DESCRIPCIÓN	
Una de las claves de éxito en un Proyecto Informático es tener requerimientos documentados, completos y aprobados por el cliente. Es extraño que se comience un proyecto informático sin requerimientos, pero es común que habiendo manifestado el cliente su necesidad, se entienda que dicha manifestación alcanza y a partir de allí comienza la interpretación de los requerimientos del cliente que seguramente comienzan a divergir de la necesidad real.	
EFFECTOS	RECOMENDACIÓN
<ul style="list-style-type: none"> <li>○ Alcance: El sistema que se entregue difícilmente satisfaga las necesidades del cliente</li> </ul>	<ul style="list-style-type: none"> <li>○ Si el cliente ha entregado una clara definición de requerimientos al solicitar el proyecto informático, se deberá incluir una primera etapa de revalidación de requerimientos con las distintas áreas del cliente.</li> <li>○ Si no se tienen requerimientos antes de comenzar, la primera etapa del proyecto consistirá en realizar un relevamiento de requerimientos y no se deberá terminar esta actividad hasta que todos los requerimientos queden claros y sean aceptados formalmente por el cliente.</li> </ul>

**No tener control de configuración**

DESCRIPCIÓN	
El control de configuración consiste en un proceso y usualmente alguna herramienta que permite determinar los componentes que forman parte de la solución y poder administrar el trabajo de distintos equipos sobre los mismos componentes. No tener este tipo de control implica que al momento de pasar la solución de un entorno de desarrollo a un entorno productivo no se tenga certeza de todos los componentes que se deben pasar, cuál versión de los componentes es válida cuando hay más de una y qué versión del programa fuente corresponde con el ejecutable (Código objeto) que fue utilizado en las pruebas.	

<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance: La falta de control de configuración afectará en forma directa la calidad del sistema pues es muy probable que en un entorno productivo la aplicación no se comporte de la misma manera que se haya comportado durante las pruebas de aceptación de la solución.</li> </ul>	
<b>RECOMENDACIÓN</b>	
<ul style="list-style-type: none"> <li>○ Tener un control de configuración estricto.</li> </ul>	

**No tener documentación**

<b>DESCRIPCIÓN</b>	
<p>Es común que se menosprecie a la documentación de la solución en un proyecto informático, tanto por el proveedor como por el cliente. La elaboración de la documentación no tiene mayor trascendencia para el proyecto y generalmente es delegada al personal de menor capacidad. Cuando falta tiempo es una de los elementos que se eliminan del plan o se posergan. La falta de documentación afecta a los proyectos cuando se producen bajas inesperadas en los equipos de trabajo y los reemplazos no tienen ningún elemento como para retomar el trabajo debiendo volver a comenzar con tareas que estaban avanzadas. Al cliente lo afecta durante el proyecto si se produce la necesidad de reemplazar al proveedor y el nuevo proveedor debe comenzar todo nuevamente y después de entregada la solución, cuando en un nuevo proyecto o ante la necesidad de realizar un mantenimiento, existe ignorancia total de cómo proceder ante la falta de documentación, elevando el riesgo y costo de cualquier acción posterior.</p>	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance, Tiempo, Costo: según la situación se verán afectadas cualquiera de estas variables.</li> </ul>	

**DESCRIPCIÓN**

Es común que se menosprecie a la documentación de la solución en un proyecto informático, tanto por el proveedor como por el cliente. La elaboración de la documentación no tiene mayor trascendencia para el proyecto y generalmente es delegada al personal de menor capacidad. Cuando falta tiempo es una de los elementos que se eliminan del plan o se posergan. La falta de documentación afecta a los proyectos cuando se producen bajas inesperadas en los equipos de trabajo y los reemplazos no tienen ningún elemento como para retomar el trabajo debiendo volver a comenzar con tareas que estaban avanzadas. Al cliente lo afecta durante el proyecto si se produce la necesidad de reemplazar al proveedor y el nuevo proveedor debe comenzar todo nuevamente y después de entregada la solución, cuando en un nuevo proyecto o ante la necesidad de realizar un mantenimiento, existe ignorancia total de cómo proceder ante la falta de documentación, elevando el riesgo y costo de cualquier acción posterior.

**EFFECTOS**

- Alcance, Tiempo, Costo: según la situación se verán afectadas cualquiera de estas variables.

**RECOMENDACIÓN**

- Darle la mayor importancia a la documentación y considerar que un proyecto informático sin documentación es de alto riesgo y una solución informática sin documentación es casi inservible.

**No tener control de documentación**

<b>DESCRIPCIÓN</b>	
<p>No tener control de la documentación es casi tan inconveniente como no tener documentación. Nunca se sabrá si la información contenida en la documentación es válida y actual o peor aún, se puede llegar a considerar que la documentación es correcta cuando en realidad no está actualizada.</p>	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance, Tiempo, Costo: según la situación se verá afectada cualquiera de estas variables.</li> </ul>	

**RECOMENDACIÓN**

- Tener un repositorio común donde se guarda toda la documentación.
- Generar respaldos del repositorio de documentación.
- Utilizar herramientas de ofimática que permitan compartir documentos y se encargue de manejar las actualizaciones simultáneas.
- Llevar un registro de la documentación, definir un proceso para utilizar nombres de documentos o archivos que deje en claro de qué versión del documento se trata. También dejar información dentro de los documentos que dejen en claro la versión, fecha de actualización y qué se actualizó.

**No controlar a los proveedores**

<b>DESCRIPCIÓN</b>	
<p>En oportunidades no se ejerce ningún control sobre los proveedores, el cliente cada tanto le pregunta cómo anda el proyecto y/o para cuándo va a estar listo. Para proyectos informáticos de envergadura esta práctica es inaceptable pues se puede descubrir demasiado tarde que el proyecto tiene por destino el fracaso.</p>	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance, Costo y Tiempo: cualquiera de estas variables puede ser impactada.</li> </ul>	

**DESCRIPCIÓN**

En oportunidades no se ejerce ningún control sobre los proveedores, el cliente cada tanto le pregunta cómo anda el proyecto y/o para cuándo va a estar listo. Para proyectos informáticos de envergadura esta práctica es inaceptable pues se puede descubrir demasiado tarde que el proyecto tiene por destino el fracaso.

**RECOMENDACIÓN**

- Exigir dentro de las condiciones del proyecto que se mantengan reuniones de control de avance periódicas en las cuales se pueda ver el plan detallado del proyecto de ser necesario.
- Si el cliente tiene personal técnico, este deberá realizar revisiones de los documentos y productos intermedios que va generando el proyecto.

**No realizar pliegos técnicos con los proveedores**

<b>DESCRIPCIÓN</b>	
<p>Existen dos situaciones en este aspecto:</p> <ol style="list-style-type: none"> <li>1) cuando el contrato entre el cliente y el proveedor no tiene anexos técnicos en donde se describe con el mayor detalle posible qué es lo que se está contratando.</li> <li>2) cuando el proveedor principal o integrador, subcontrata partes del proyecto con otros proveedores y en el contrato correspondiente no se detalla en forma minuciosa el servicio que se está solicitando.</li> </ol>	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Alcance: la funcionalidad que provea el proyecto podrá estar incompleta.</li> </ul>	

**RECOMENDACIÓN**

- Deben existir anexos técnicos en los distintos contratos que especifiquen en forma detallada cada uno de los servicios o productos a proveer.

**Convención de nombres**

<b>DESCRIPCIÓN</b>	
<p>Cuando se desarrolla una solución en un proyecto informático, que se debe instalar en un entorno técnico existente, se deben tener en cuenta las convenciones de nombres a ser utilizados. No hacerlo en ocasiones puede imposibilitar la instalación de una solución. Esta situación es común cuando la solución se desarrolla y prueba en un entorno del proveedor y cuando se la traslada al entorno del cliente comienzan a surgir los problemas.</p>	
<b>EFFECTOS</b>	
<ul style="list-style-type: none"> <li>○ Tiempo: No tener esto en cuenta posiblemente produzca demoras en el momento de tener que instalar la aplicación.</li> </ul>	

**RECOMENDACIÓN**

- El equipo del proveedor deberá consultar con el responsable del entorno técnico del cliente, si usan convenciones de nombres en la instalación y en caso negativo igualmente solicitar los nombres utilizados dentro del entorno en que se instalará la solución a desarrollar.

**Estándares de programación**

DESCRIPCIÓN	
El proveedor deberá consultar con el cliente si utiliza estándares de programación en sus sistemas. De ser así, deberá adoptar dichos estándares o negociar para utilizar los propios. De no hacer esta consulta de manera oportuna, puede surgir un conflicto en el momento de la aceptación del proyecto.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: Se producirán demoras si el proveedor debiese reconstruir los programas mal codificados.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ El proveedor deberá consultar con el cliente si utiliza estándares de programación en sus sistemas.</li> </ul>	

**Técnicas de análisis, diseño y pruebas**

DESCRIPCIÓN	
No utilizar técnicas de análisis, diseño y pruebas es un indicio de falta de madurez de parte del proveedor a cargo de un proyecto informático de envergadura. Tener definidas técnicas estándar y utilizarlas en los proyectos garantizará que el trabajo se realice de una manera más homogénea y eficiente. Permitirá reaprovechar el trabajo realizado en futuros proyectos que utilicen las mismas técnicas. Un equipo de proyecto en donde cada cual hace las cosas como más le parece, estará utilizando la receta para el caos.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: tarde o temprano habrá que rehacer trabajo y esto generará demoras.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ El cliente debería preguntarle al proveedor si utiliza técnicas estándar dentro de sus equipos, de análisis, diseño y pruebas. Si la respuesta es afirmativa debería solicitarle que enumere dichas técnicas para evaluar (o consultar) si dichas técnicas son adecuadas.</li> </ul>	

**Falta de modelo de datos**

DESCRIPCIÓN	
La existencia de un modelo de datos facilita el trabajo en un proyecto, especialmente si está bien documentado y comunicado, pero más importante aún, significará que alguien pensó y diseñó dicho modelo. No contar con un modelo de datos implica que posiblemente se genere cierto caos en la construcción de la solución y que sea muy difícil mantener más adelante el sistema.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Alcance: la calidad del sistema se verá seriamente afectada si no se ha utilizado un modelo de datos.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Exigir que se construya la solución en base a un modelo de datos y que exista un proceso y control que asegure que todo el desarrollo respeta dicho modelo.</li> </ul>	

**Comenzar sin acuerdo formal**

DESCRIPCIÓN	
Esta situación es demasiado común. Los informáticos son muy ansiosos por empezar y desarrollar las brillantes soluciones que han pensado y los comerciales son muy propensos a satisfacer la urgencia del cliente. Generalmente se llega a un entendimiento sobre el proyecto que se necesita y se comienza con la promesa que la firma de contrato estará lista muy pronto. Es común que luego la firma se demore, que la gerencia de administración y finanzas no esté enterada del proyecto y que debe pagar facturas y en determinado momento se produce un quiebre.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Más que un impacto al Alcance, Tiempo y Costo, se produce una crisis que más de las veces termina en una cancelación y un juicio.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Ser totalmente riguroso en no comenzar con la ejecución de un proyecto hasta no tener todo acordado, escrito y firmado.</li> </ul>	

**No comenzar una etapa sin terminar la anterior**

DESCRIPCIÓN	
En los ciclos de desarrollo denominados cascada, se supone que se termina una etapa para luego comenzar con la próxima. Esto es así porque en base a los requerimientos se realiza el Diseño y en base al Diseño se realiza la Construcción. Si en cada etapa se gestiona una aprobación parcial, el comenzar con la etapa siguiente antes de finalizar la etapa en curso significa que quizás surja un cambio que haga rehacer cosas que ya se comenzaron de la etapa siguiente, o que el cliente no me acepta algo y/o introduce un cambio inesperado que hace inválido el trabajo que se adelantó y que no se puede recuperar.	
EFFECTOS	
<ul style="list-style-type: none"> <li>○ Tiempo: rehacer trabajo que se adelantó puede terminar generando demoras.</li> </ul>	
RECOMENDACIÓN	
<ul style="list-style-type: none"> <li>○ Si el ciclo de desarrollo exige un proceso en cascada, se deberá respetar al máximo y evitar adelantar trabajo de la etapa siguiente.</li> </ul>	

**5. CONCLUSIONES****ACCIONES POSIBLES**

Ante la gran cantidad de problemas detallados se pueden realizar dos acciones claramente contrapuestas como son la Prevención y la Remediación.

**PREVENCIÓN**

Como seguramente el lector pudo apreciar, muchos de los problemas mencionados en la sección anterior son muy simples e inexplicablemente demasiado usuales. Las malas prácticas continúan como informa el Standish Group, pues no es posible llegar a tener ni un tercio de los proyectos informáticos exitosos. Como sugiere el autor a través de la enumeración de problemas, el origen de dichos problemas se concentra en solo dos áreas y por lo tanto las acciones de prevención deberían ir dirigidas sobre estas áreas:

1. Contratos y anexos técnicos.
2. Metodología.

Los contratos informáticos detallan el acuerdo entre las partes y pasan a ser una de las principales pruebas ante una eventual demanda. Por tal motivo hay que poner especial énfasis en la preparación y negociación de un contrato. Se recomienda tener en cuenta los problemas mencionados y guías como la que proveen Roberto ALFARO PINILLOS y Jorge VELARDE SUSSON<sup>9</sup> con los ocho pasos para negociar un contrato de software:

1. Empezar temprano;
2. Mantener el control;
3. Identificar los malentendidos;
4. Documentar las especificaciones;
5. Insistir en pruebas para recibir satisfacción;
6. Prever un desempeño continuado;
7. Atar los pagos a entregas; y
8. Redactar simplificando los datos técnicos.

Obviamente no todos los problemas tienen la misma probabilidad de generar conflictos, así que a modo de resumen podemos decir en base al Standish Group y la experiencia del autor que los factores que más contribuyen a conflictos son:

1. Falta de participación del usuario y/o falta de apoyo de los ejecutivos.
2. Falta de requerimientos claros y/o requerimientos cambiantes.
3. Deficiencias en la redacción de los contratos informáticos.

#### Remediaciόn

Si no se adoptaron las medidas de prevención sugeridas o no alcanzaron, se llegará a tener que manejar una crisis o ruptura en el proyecto, ante lo cual existirán dos caminos a seguir:

- Negociación o mediación
- Litigación y peritaje

La negociación es el camino más recomendado y menos oneroso. Se pueden hacer participar a árbitros calificados y a peritos que den una visión imparcial del proyecto y las cuestiones que hayan generado el conflicto.

Las demandas son onerosas, llevan tiempo y no siempre nos terminan dando una satisfacción. Podemos descubrir que las mismas deficiencias contractuales juegan en contra al momento de aportar pruebas y las decisiones de peritos oficiales no siempre coincidirán con la de nuestro perito de parte y no por una cuestión de parcialidad sino porque habrá temas de interpretación y de experiencia que producirán informes distintos.

En otras palabras sucede lo mismo que con la calidad del software esto es el costo de resolver un error en las etapas tempranas es muy bajo, pero se torna excesivamente costoso resolver el mismo error cuando el software se encuentra operativo.

Por lo tanto es recomendable invertir en prevención, contratar abogados y expertos informáticos al momento de redactar los contratos y seleccionar proveedores que demuestren utilizar una metodología robusta y tengan éxitos comprobables en sus antecedentes.

## CONCLUSIONES FINALES

Hemos visto cómo una rama moderna de la Ingeniería como la Informática, ha realizado esfuerzos por alcanzar un nivel de calidad y confiabilidad similar a las otras ramas. No obstante la formación de los profesionales informáticos en aspectos metodológicos de desarrollo de software y desarrollo de proyectos informáticos, aún en general aparece como complementaria a las formaciones de grado. Por lo tanto la oferta de servicios informáticos aún es brindada por una amplia mayoría de profesionales, técnicos e inclusive personas sin formación formal, que carecen de formación metodológica apropiada. Seguramente esto irá cambiando lentamente en el tiempo en la medida que se mejoren los planes de estudio y se comience a exigir niveles de certificación metodológica a los profesionales del ramo.

Mientras tanto las estadísticas de proyectos informáticos seguirán mostrando alta cantidad de proyectos no exitosos ante los cuales se podrán tener actitudes reactivas o proactivas por parte de las empresas que contraten servicios informáticos. Lamentablemente las empresas terminan aplicando por lo general medidas reactivas pues salvo excepciones pasan a ser el contratante débil debido a la brecha de conocimiento técnico en relación al proveedor del servicio, por la falta de experiencia en la contratación de este tipo de servicios y la falta de conocimiento sobre temas informáticos de parte de los asesores legales de las empresas.

Es por lo tanto una responsabilidad ética profesional de los informáticos el divulgar la necesidad de emplear metodologías de proyecto y desarrollo de software que aseguren un proceso de calidad.

El autor espera haber realizado con esta monografía una pequeña contribución en tal sentido, tras resumir una cantidad de errores que por lo general generan conflictos y brindar recomendaciones de cómo evitarlos, para un espectro reducido de los proyectos informáticos como lo son los proyectos de envergadura de integración de sistemas pero que no obstante reflejan cómo se realizan en general los proyectos informáticos.

Para quienes deseen profundizar en los temas metodológicos se los invita a realizar búsquedas en Internet sobre temas tales como:

- Ingeniería de Software y Madurez de los Procesos de desarrollo del Software, buscando la sigla CMM<sup>10</sup>,
- Gerenciamiento de Proyectos o Project Management, buscando la sigla PMI<sup>11</sup>,
- Procesos de calidad para lo cual puede buscar las siglas ISO 9000<sup>12</sup> y SIX Sigma<sup>13</sup>.

<sup>10</sup> Ingeniería de Software y Madurez de Procesos de desarrollo de Software = [www.sei.cmu.edu/cmm/](http://www.sei.cmu.edu/cmm/)

<sup>11</sup> Gerenciamiento de Proyectos = [www.pmi.org](http://www.pmi.org) [www.pmi.cl](http://www.pmi.cl)

<sup>12</sup> Calidad = [www.iso.org](http://www.iso.org)

<sup>13</sup> Calidad = [www.ge.com/sixsigma](http://www.ge.com/sixsigma)

<sup>9</sup> Compendio Práctico de Contratos Civiles, Comerciales, Bancarios e Informáticos, editorial San Marcos (España), 2000, pp. 216.

Con estas búsquedas podrán identificar las organizaciones que impulsan procesos de calidad, firmas que brindan soporte en dichas áreas, más bibliografía que el lector puede consultar y noticias sobre casos de éxito<sup>14</sup>.

<sup>14</sup> Ejemplos sobre éxitos locales, [Revista Gerencia](#).

"Les cuento que Altec Chile se ha convertido en la primera empresa de software de Chile y la cuarta de Iberoamérica en recibir la evaluación CMMI Nivel 5 ... [Empresas de desarrollo de software logran certificación ISO 9001:2000](#)

Ocho empresas chilenas de desarrollo de software confirmaron la importancia de asociarse para fortalecer la calidad de sus procesos y productos.