Centro de Investigación en Matemáticas, A. C.

Aplicando TRIZ en el Proceso de Desarrollo de Arquitecturas de Software

Reporte Técnico de Investigación
para obtener el grado de

Maestro en Ingeniería de Software

presenta

Mery Helen Pesantes Espinoza

Asesor:
Dr. Cuauhtémoc Lemus Olalde

Guanajuato, Gto. 28 de Agosto de 2006.

A mi esposo José Arturo
por su constante apoyo y ánimo.
Gracias

AGRADECIMIENTOS

Al Dr. Cuauhtémoc Lemus Olalde, mi asesor en este reporte técnico, por su tiempo para dirigirme en mi investigación y por la oportunidad de ser parte de esta segunda generación.
Al Dr. Carlos Montes de Oca Vásquez, por su entera disposición y guía durante este año y por la oportunidad de ser parte de esta segunda generación.
Al Consejo de Ciencia y Tecnología del Estado de Guanajuato (CONCYTEG) a través del proyecto GTO-2002-C01-5333 titulado "Promoviendo calidad en la industria de software: recursos humanos, investigación, servicios" por apoyarme con una beca para la realización del estudio de una maestría de calidad.
A mis amigos y compañeros de la segunda generación: Araceli Nuñez, Dulce María Gomez, Edwin Gutierrez, Giberto Grajales y Cesar Collí, por su organización personal y su disciplina, aprendí mucho, muchas gracias.

TABLA DE CONTENIDO

LISTA DE FIGURAS

APÉNDICES

1 Introducción

La competencia entre tecnologías y servicios esta aumentando cada vez más, esta es una era de crisis y aún una era de oportunidades, solamente quienes puedan resolver problemas creando nuevas ideas e implementándolas rápidamente podrán sobrevivir. La habilidad de crear para individuos y la capacidad de resolver problemas para organizaciones, son capacidades fundamentalmente basadas en la mente humana, incluso tecnólogos e investigadores educados en varias especialidades no son siempre bastantes creativos. Entonces ¿cómo nosotros podemos incrementar nuestra creatividad?. Recientemente, TRIZ ha destacado por tener la posibilidad de cambiar dramáticamente esta situación.

TRIZ es una abreviación rusa que puede ser traducido como “Teoría de la Solución Inventiva del Problema” creado por Genrich Altshuller, quien estuvo interesado en basar TRIZ sobre la ciencia. Él comenzó a desarrollar TRIZ en 1946, pero es hasta 1956 que su primer artículo es publicado en Rusia. TRIZ llega a ser una metodología poderosa para la creatividad en diferentes campos de la ingeniería, era desconocido fuera de Rusia hasta 1990 y su popularidad en USA, Japón y Europa está creciendo rápidamente y muchas compañías han citado un fenómeno de incremento de su productividad al incorporarlo por lo que están financiando TRIZ por sus ideas y soluciones de calidad a problemas difíciles de ingeniería.

Considerando el desarrollo que está teniendo TRIZ, han surgido diferentes ramas de aplicación, tales como: TRIZ con QFD, TRIZ basado en Function-Cost Analysis, TRIZ con Value Engineering, TRIZ y Pedagogía y recientemente se están realizando investigaciones para aplicar TRIZ a Software. Por lo que este trabajo está orientado a estimular la aplicación de TRIZ a los problemas de dominio de software, principalmente en la etapa de diseño de software.

Como es sabido la base de cualquier sistema de software es su arquitectura y por años los diseñadores de software han estado desarrollando sistemas sin tomar en cuenta este aspecto y por lo tanto los sistemas resultantes no son lo que los clientes desean ó en el mejor de los casos no cumple con todos los requerimientos establecidos, por lo tanto se realizan sistemas sin calidad y se requiere de más tiempo y esfuerzo para repararlos. Entonces, la arquitectura de software ha emergido como una parte muy importante en el proceso de diseño y es el objeto de este trabajo integrar TRIZ sobre el proceso de desarrollo de arquitecturas de software basado en DFSS para asegurar que nuevos productos son diseñados reuniendo altos niveles de calidad, para resolver contradicciones en los requerimientos de diseño, acortar el tiempo de desarrollo de una nueva arquitectura, reducir errores, disminuir costos, mejorar la satisfacción del cliente produciendo productos amigables con el usuario. Además, de permitir a los estudiantes, Ingenieros de Software y Organizaciones a estimular su pensamiento innovador para resolver problemas difíciles relacionados al diseño de una nueva arquitectura.

El presente trabajo se encuentra estructurado para un buen entendimiento de la siguiente manera:
Parte 1: **Introducción**.- Describe de manera resumida el panorama general a tratar en el presente trabajo.
Parte 2: Antecedentes de TRIZ.- se menciona su historia, definición, fuentes de origen, herramientas/métodos, ramas de aplicación y se detalla el proceso general que utiliza para solucionar problemas.
Parte 3: Trabajo previo.- Describe los trabajo previos encontrados de aplicaciones TRIZ en otros campos.
Parte 4: Descripción del problema.- Describe cual es la problemática a la que se enfrenta la Ingeniera de Software en el área de diseño de arquitecturas de software.

Parte 5: Propuesta de solución.- Describe la propuesta de solución a la problemática presentada de cómo TRIZ puede ayudar en el diseño de arquitecturas de software, presentando un mapeo de las metodologías utilizadas (TRIZ y el proceso de desarrollo de arquitecturas de software).
Parte 6: Conclusiones y trabajos futuros.- Describe las conclusiones finales del trabajo y los temas que aún necesitan mas investigación que podrían derivar nuevos trabajos.
Parte 7: Glosario.- Describe los conceptos utilizados en el presente trabajo con la finalidad de tener un mejor entendimiento del tema.
Parte 8: Acrónimos.- Describe los acrónimos utilizados en el presente trabajo con la finalidad de tener un mejor entendimiento del tema.
Parte 9: Referencias.- Describe las referencias bibliográficas que en las que está basado el presente trabajo de investigación, tales como: libros, artículos y páginas web.
Parte 10: Apéndices.- Se muestran los materiales auxiliares para la aplicación adecuada de TRIZ, tales como: parámetros, principios y la matriz de contradicción de TRIZ

2 Antecedentes de TRIZ (Teoría de la Solución Inventiva del Problema)

2.1 Historia

TRIZ, es una acrónimo ruso para “Teoriya Resheniya Izobretatelskikh Zadatch” y traducido al inglés como “Theory of Inventive Problem Solving (TIPS)”, desarrollada por Genrich Saulovich Altshuller(1926 -1998) y sus colegas en la ex Unión Soviética en 1946.
Genrich Altshuller obtuvo la idea inicial de TRIZ cuando estuvo trabajando en la oficina de patentes de la Marina de Guerra, creyó que era posible aprender a ser un inventor, por lo que para probar esto desarrolló la teoría y la práctica de la ciencia de la invención. En sus etapas iniciales de trabajo reconoció que entre un gran número de patentes existían ideas similares y soluciones análogas en diferentes áreas y para diferentes problemas, dándose cuenta que incluso inventos creativos y originales naturalmente tenían patrones comunes. De esa manera él dijo que deberíamos extraer los patrones de invención de las mejores patentes y estudiarlas.
Altshuller escribió una carta a Stalin (Gobernador Ruso) sobre sus ideas como una propuesta, quien asume que estaba contra el régimen y lo encarceló bajo cargos políticos por cinco años. Pero Altshuller continuó con su trabajo desarrollando: 40 principios de la innovación, varias leyes de la evolución de sistemas técnicos, conceptos de contradicciones físicas y técnicas, el concepto de idealidad de un sistema entre otros enfoques teóricos y prácticos, este amplio trabajo representa una única contribución al desarrollo de la creatividad y para la solución de problemas de inventiva.

Las investigaciones en esta disciplina iniciaron bajo la hipótesis de la existencia de principios universales para la invención. Estos principios son la base de las innovaciones creativas que permiten el avance tecnológico de la humanidad y si esos principios pueden ser descubiertos y codificados, estos pueden transmitirse a las personas y permitir que el proceso de invención ser mas predecible.
La investigación continuó en varios niveles a lo largo de casi 60 años, desde entonces más de 2 millones de patentes han sido examinadas, clasificadas y analizadas en búsqueda de dichos principios de innovación, obteniendo los siguientes postulados:

- Los problemas y soluciones tienden a repetirse a través de la industria y ciencias.
- De igual manera los patrones de evolución tecnológica se repiten a través de varias industrias y ciencias.
- Los efectos científicos de estas funciones pueden utilizarse en otros campos.

Mientras Genrich Altshuller estuvo vivo repetidamente dijo que “TRIZ como una ciencia debe pertenecer a todo” diciendo que TRIZ no tendría limitaciones para quién y donde lo puedan utilizar.

En el pasado, los gestores de los proyectos habían evitado concientemente todos los tipos de creatividad, debido a que creían que la solución creativa de problemas incrementaba el riesgo de que el proyecto fracasara, desde entonces la creatividad ha tenido la reputación de no ser la adecuada para un proceso formal de desarrollo, debido a la indisciplinada generación de nuevas ideas, misma que limitaban el sentido práctico, así que suponían que era un peligro al proyecto. TRIZ es un método de solución de problemas que acelera la habilidad del equipo del proyecto para resolver esos riesgos generadores de problemas [17].
Se considera una ciencia internacional para la creatividad, que se fundamenta en el estudio de patrones de problemas y soluciones que reduzcan la brecha hacia la solución de problemas técnicos, no se basa en la creatividad espontánea de individuos o grupos, ya que gran parte de la práctica de TRIZ consiste en aprender esos patrones de problemas - soluciones repetidas, patrones de evolución técnica, así como métodos usados en los efectos científicos y aplicar los patrones de TRIZ a la situación específica que enfrentamos.

TRIZ no es el primer intento para establecer principios científicos de innovación, el método "And – And" por R.L. Bartini es veinte años mas antiguo que TRIZ, ambos métodos nacen de la lógica dialéctica de manera independientemente y en diferentes tiempos.

En 1990, asociaciones y seguidores de Altshuller fueron enseñando TRIZ aproximadamente en 200 escuelas a siete mil estudiantes. Como resultado de la caída de Rusia y el subsecuente fin de la Guerra Fría, un gran número de especialistas de TRIZ tuvieron que emigrar a USA y Europa difundiendo la metodología. En Japón TRIZ fue introducido en 1996 y tuvo más acogida en las industrias, especialmente en manufactura.

2.2 Definición

TRIZ es una metodología sistemática de solución inventiva de problemas basada en el conocimiento orientado al ser humano [1].

Conocimiento.- TRIZ puede definirse como un enfoque basado en el conocimiento porque:

a) El conocimiento acerca de soluciones heurísticas de problemas genéricos es extraído de un gran número de patentes a nivel mundial en diferentes campos de la ingeniería. TRIZ argumenta que este trabajo con un número relativamente pequeño de objetivos heurísticos está basado sobre las tendencias de evolución de técnicas. Esta declaración no está probada y está basada solamente sobre un análisis estadístico de soluciones representadas en patentes.
b) Este usa el conocimiento de los efectos de la ciencia natural y en la ingeniería. Este gran almacén de información es resumida y organizada para un eficiente uso durante la solución de problemas.
c) Este usa el conocimiento acerca del dominio donde el problema ocurre. Este conocimiento incluye información acerca de la misma técnica, tanto como sistemas y procesos similares u opuestos, técnicas del entorno y su evolución o desarrollo.

Orientado al ser humano.- la heurística esta orientada a ser usada por un ser humano y no por una máquina. La práctica de TRIZ está basada sobre la división de una técnica en subsistemas, distinguiendo sus funciones útiles y dañinas. Tales operaciones son arbitrarias, porque ellos dependen sobre el problema en sí y sobre circunstancias socio económicas, además que ellas no pueden ser realizadas por una computadora. Para la mayoría de los problemas que se muestran, los cuales son repetidos una y otra vez para esto es razonable usar computadoras. Sin embargo, muchos problemas ocurren solamente una vez y para esto no es eficaz usar computadoras, nosotros necesitaremos invertir más tiempo escribiendo programas que serían requeridos para que una persona resuelva un problema técnico. Por consiguiente, nosotros necesitamos armar un solucionador humano con un instrumento para manejar tales problemas.

Sistemático.- en la definición de TRIZ tiene dos significados:

a) Modelos genéricos y detallados de sistemas artificiales y procesos son considerados dentro del framework de análisis especial de TRIZ y el conocimiento sistemático u organizado acerca de la importancia de estos procesos y sistemas.
b) Procedimientos para resolver problemas y la heurística son estructurados sistemáticamente para proporcionar aplicaciones efectivas de soluciones conocidas para nuevos problemas.

Problemas de Invención y Soluciones.- Principales abstracciones para soluciones inventivas de problemas incluye:

- Frecuentemente pasos desconocidos aparecen debido a los requerimientos contradictorios para el sistema.
- Frecuentemente situaciones deseables desconocidas pueden ser reemplazadas temporalmente por una solución ideal imaginaria.
- Usualmente la solución ideal puede ser obtenida debido a recursos de un entorno o dentro de la técnica.

- Usualmente la solución ideal puede ser proyectado de evolución técnica de tendencias conocidas.

2.3 Fuentes

Las principales fuentes de TRIZ han sido las patentes y la información técnica. Los especialistas de TRIZ han analizado aproximadamente 2 millones de patentes a nivel mundial que representa aproximadamente el 10% de todas las patentes en el mundo. Este amplio e increíble trabajo fue posible por los siguientes factores:

- Ingenieros en la antigua Unión Soviética estuvieron invirtiendo 8 horas en su lugar de trabajo (su salario no dependía de su esfuerzo, experiencia o cantidad y calidad del trabajo), muchos de ellos influenciados por Altshuller y TRIZ usaron este tiempo para estudiar patentes. Ejemplo: Altshuller estudió más de 400,000 patentes a nivel mundial para crear la matriz de contradicción con 40 principios inventivos.
- Patentes rusas tenían una estructura simple: resumen, cuerpo de la patente y pocas argumentaciones. Resúmenes de todas las patentes de diferentes países (USA, Japón, Europa, Francia, etc) fueron traducidas a ruso.
- Varios departamentos federales de Rusia, cubriendo cada rama de la industria, publicaron revisiones periódicas de los reportes técnicos más importantes y publicaciones científicas en numerosos campos.
- El análisis de las patentes siguieron un buen procedimiento, donde se clasifican las patentes en cinco niveles.

El principal resultado de estos estudios es la colección de heurísticas de TRIZ que ayudan a resolver problemas no rutinarios. Frecuentemente una heurística de TRIZ puede ayudar a realizar los pasos más importantes en la solución de un problema, llenando la brecha entre la situación inicial y lo deseado. Si una secuencia de pasos es necesaria para resolver problemas, TRIZ ofrece poderosos instrumentos para manejar diferentes aspectos de la solución de problemas.

2.4 Herramientas/Métodos

- **Análisis Preliminar**.- puede evitar soluciones trade-off de problemas que contienen contradicciones y puede ayudar a aclarar información importante acerca de la técnica y restricciones de soluciones futuras.
- **Matriz de contradicción**.- consiste de contradicciones técnicas entre las características a ser mejoradas y las características que pueden adversamente ser afectadas. Esto también tiene algunos principios inventivos en cada celda que puede ayudar a resolver las contradicciones.
- **Principios de separación**.- Ayuda a resolver contradicciones físicas entre las características opuestas de un simple subsistema.
- **Substance-Field (Su-Field).-** es un enfoque modelado basado sobre el lenguaje simbólico que puede registrar transformaciones de sistemas técnicos y procesos tecnológicos.
- **Enfoques estándares para problemas de invención (estándares).-** está basado sobre la observación de muchos problemas técnicos de inventiva en diferentes campos de la ingeniería que son resueltos con los mismo enfoques genéricos. Los estándares contienen típicos problemas inventivos y recomendaciones típicas de soluciones, lo cual usualmente puede ser presentado en términos de análisis de Su-Field.
- **Algoritmo para soluciones inventivas de problemas (acrónimo ruso ARIZ).-** es un conjunto de procedimientos lógicos secuenciales para eliminar contradicciones que causan el problema. ARIZ es considerado como uno de los instrumentos más poderosos y elegantes de TRIZ. Este incluye el proceso de reformulación y reinterpretación del problema hasta alcanzar la definición precisa y el proceso disciplinado y lógico de resolver problemas con un uso iterativo de las heurísticas de TRIZ.
- **Método agente.-** es un procedimiento gráfico- lógico implementando pasos hacia delante, atrás o bidireccional entre las situaciones iniciales y deseada, cuando ellos

puedan ser representados con una declaración correcta de un problema y el Resultado Final Ideal.

- **Directed Product Evolution (DPE).-** Trata de predecir las características futuras de una máquina, procedimiento o técnica, se basan en simulaciones, evaluaciones y tendencias para crear un modelo probabilístico de futuro desarrollo.

2.5 Ramas

TRIZ ha sido desarrollado exclusivamente en la antigua Unión Soviética por un par de docenas de TRIZniks (especialistas en TRIZ). Muchos TRIZniks también enseñaron TRIZ a miles de ingenieros que han llegado a familiarizarse con esta metodología, TRIZ llegó a ser tan popular como el método de Lluvia de Ideas fue en los países del occidente.
TRIZ era desconocido fuera de Rusia debido a las barreras de idioma y política. Sin embargo durante los últimos 50 años pocos métodos han sido propuestos e implementados en USA, Europa y Japón tales como: Value Engineering, Design Theories, Quality Function Deployment (QFD), Taguchi Method y Theory of Constraints.
Algunos proyectos de TRIZ están basados en la reingeniería creativa de ideas particulares que han sido desarrollados en el framework de estos métodos, Tal desarrollo guía a la creación de nuevas ramas de TRIZ tales como TRIZ basado en Function-Cost Analysis ó TRIZ + QFD.
A fines de 1980, TRIZ llega a ser una poderosa herramienta siendo aplicado en un amplio rango de problemas no técnicos, tales como administración, educación, relaciones públicas e inversión (denominada aplicaciones “soft”).

2.6 Proceso

Altshuller después de analizar 1,500,000 patentes e identificar las soluciones inventivas las categorizó de una manera novedosa. El encontró que los problemas habían sido resueltos una y otra vez utilizando cuarenta principios inventivos fundamentales y si los inventores posteriores hubieran conocido estos trabajos, las soluciones hubieran sido resueltas rápida y eficazmente.
En 1960 y 1970, Altshuller reconoció las soluciones sobre 5 niveles, las cuales son mostradas en la siguiente figura:

Level	Degree of inventiveness	% of solutions	Source of knowledge
1	Apparent solution	32%	Personal knowledge
2	Minor improvement	45%	Knowledge within company
3	Major improvement	18%	Knowledge within the industry
4	New concept	4%	Knowledge outside the industry
5	Discovery	1%	All that is knowable

Figura 2.1. Categorización de Soluciones por Altshuller [19]

- Nivel 1.- Problemas de diseño rutinarios resueltos con métodos conocidos dentro de la especialidad. No necesita invención. Cerca del 32% de las soluciones caen en este nivel.
- Nivel 2.- Mejoras menores a un sistema existente por métodos conocidos dentro de la industria. Cerca del 45% de las soluciones caen en este nivel.
- Nivel 3.-Mejoras fundamentales sobre un sistema existente, por métodos conocidos fuera de la industria. Cerca del 18% de las soluciones caen en esta categoría.
- Nivel 4.- Una nueva generación que utiliza nuevos principios para realizar funciones principales del sistema. Soluciones fueron encontradas mas en ciencia que en tecnología. Cerca del 4% de las soluciones caen sobre esta categoría.
- Nivel 5.- Un descubrimiento científico raro ó innovación pionera de un sistema nuevo y esencial. Cerca del 1% caen en esta categoría.

2.6.1 Proceso TRIZ: Paso por Paso

Como se mencionó anteriormente, Altshuller cae en una teoría de invención aceptable, que debería ser lo suficientemente familiar para los inventores continúen el enfoque de resolver problemas mostrado en la Figura 2.2.

Los pasos identificados para la solución de problemas de nivel 1 y 2 son:

1. Identificando el Problema.- Boris Zlotin and Alla Zusman, principales científicos de TRIZ y estudiantes de Altshuller desarrollaron un “Cuestionario Innovativo de Situaciones”, para identificar sistemas de ingeniería que están siendo estudiado, entorno de operación, requerimiento de recursos, funciones principales, efectos dañinos y el resultado ideal.
2. Formula el Problema.- El Prisma de Triz.- Decir el problema en términos de contradicciones físicas(Ver **Apéndice C**), identifica problemas que podrían ocurrir, las preguntas que pueden ayudar a formular el problema son:
   - Podría mejorar una característica técnica para resolver un problema lo cual causa otras características técnicas dañinas?
   - Existen conflictos técnicos que pueden forzar un trade-off?
3. Búsqueda de problemas bien resueltos previamente.- Altshuller extrajo de 1, 500, 000 patentes a nivel mundial 39 estándares de características técnicas que causan conflicto, estos son llamados 39 parámetros de ingeniería (Ver **Apéndice B)**. Primero encuentra los principios que necesitas cambiar, entonces encuentra efectos secundarios no deseados.
4. Busca soluciones análogas y adáptalo a la solución.- también Altshuller extrajo del mundo 40 principios inventivos.

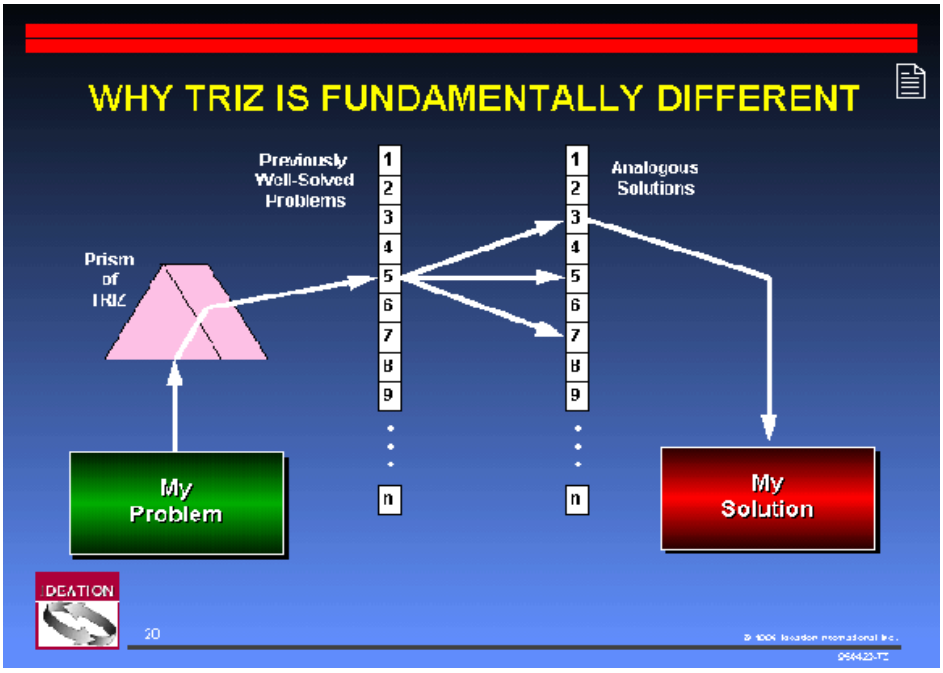

Figura 2.2. Proceso General para Solucionar Problemas [19]

Ejemplo de Aplicación: Air Bag

1. Considerar el propósito de cambiar la velocidad de inflar una bolsa de aire, para reducir los daños de los pasajeros.
2. El trade-off es que los daños en accidentes por alta velocidad incrementan.
3. Los parámetros en la matriz de contradicción serían: "Duration de action of a moving object" (Fila 15- parémetro para mejorar) y "object-generated harmful effects" (columna 31-parámetro negativo)

	Worsening Feature → / Improving Feature ↓	Volume of moving object	Speed	Force (Intensity)	Stress or pressure	Shape	Reliability	Object-generated harmful factors	Ease of operation	Ease of repair	Device complexity	Difficulty of detecting and measuring
		7	9	10	11	12	27	31	33	34	36	37
9	Speed	7, 29, 34	+	13, 28, 15, 19	6, 18, 38, 40	35, 15, 18, 34	11, 35, 27, 28	2, 24, 35, 21	32, 28, 13, 12	34, 2, 28, 27	10, 28, 4, 34	3, 34, 27, 16
10	Force (Intensity)	15, 9, 12, 37	13, 28, 15, 12	+	18, 21, 11	10, 35, 40, 34	3, 35, 13, 21	13, 3, 36, 24	1, 28, 3, 25	15, 1, 11	26, 35, 10, 18	36, 37, 10, 19
11	Stress or pressure	6, 35, 10	6, 35, 36	36, 35, 21	+	35, 4, 15, 10	10, 13, 19, 35	2, 33, 27, 18	11	2	19, 1, 35	2, 36, 37
12	Shape	14, 4, 15, 22	35, 15, 34, 18	35, 10, 37, 40	34, 15, 10, 14	+	10, 40, 16	35, 1	32, 15, 26	2, 13, 1	16, 29, 1, 28	15, 13, 39
15	Duration of action of moving object	10, 2, 19, 30	3, 35, 5	19, 2, 16	19, 3, 27	14, 26, 28, 25	11, 2, 13	21, 39, 16, 22	12, 27	29, 10, 27	10, 4, 29, 15	19, 29, 39, 35
33	Ease of operation	1, 16, 35, 15	18, 13, 34	28, 13 35	2, 32, 12	15, 34, 29, 28	17, 27, 8, 40		+	12, 26, 1, 32	32, 26, 12, 17	

Figura 2.3. Matriz de Contradicción para solucionar ejemplo [17]

Se consideran para hallar la solución los principios inventivos “21, 39, 16, 22”:
Principio 22: Skipping.- Conducir un proceso o ciertas etapas (Ejemplo: destructible, operaciones dañinas o peligrosas) en alta velocidad.

- Inflar la bolsa de aire lo más rápido que en la práctica actual, para que esté totalmente inflada cuando impacten a la persona.
- Pequeñas bolsas alcanzarían a inflarse rápidamente, así el pasajero estaría protegido en el accidente.

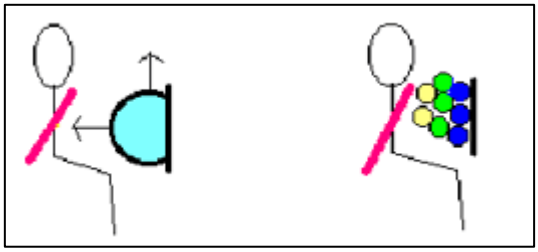

Figura 2.4. Ideas de Solución basadas en los Principios Inventivos de TRIZ [17]

2.6.2 Herramientas adicionales de TRIZ

La metodología TRIZ puede ser adaptada para resolver diferentes tipos de problemas, el método mencionado anteriormente es relativamente fácil pero fuerza al usuario pre formular el problema en términos de parámetros estándar de ingeniería, lo que rara vez te guía a un conjunto exhaustivo de soluciones. Por lo que es usado para resolver problemas de nivel 1 y 2 tal como se muestra en la Figura 2.1. Para problemas mas difíciles pueden ser resueltos con las siguientes herramientas:

1. ARIZ (Algortithm for inventive problem solving): que consta de los siguientes pasos:
    - Reestructuración del problema original:
        - Analiza el sistema.
        - Analiza los recursos.
        - Define el resultado final y formula la contradicción física.
    - Removiendo la contradicción física:
        - Separa la contradicción física.
        - Aplica el conocimiento base: efectos, estándares y principios.
        - Cambio al Mini-problema.
    - Analizando la solución:
        - Revisa la solución y analiza el remover contradicciones.
        - Desarrolla el uso máximo de la solución.
        - Revisa todas las etapas en ARIZ en aplicaciones de tiempo real.
2. Substance - Field Analysis:
    - Los objetos son llamados substancias y la acción un campo.
    - Es útil para identificar fallas funcionales (acciones no deseadas que pueden aplicarse en un campo).
3. Anticipatory Failure Determination (AFD):
    - Previene fallas imprevistas en un nuevo producto desarrollado.
    - Inventa mecanismos de fallas y entonces examina las posibilidades de su ocurrencia.
4. Directed Product Evolution (DPE):
    - Trata de predecir las características futuras de una máquina, procedimiento o técnica.
    - Se basan en simulaciones, evaluaciones y tendencias para crear un modelo probabilístico de futuro desarrollo.

2.7 Compañías Usando TRIZ

Se muestra a continuación una lista de las compañías que han comenzado a estudiar y usar TRIZ desde 1993:

- Ford Motor Co.
- General Motor Corp.
- Johnson & Johnson.
- Rockwell International.
- Xerox Corporation.
- UNISYS
- Chrysler Corp.
- Emerson Electric.

2.8 Software TRIZ

Porque TRIZ es construido sobre una base de datos de cientos de miles de patentes, principios, operadores, contradicciones, etc. utiliza software para ayudar a los ingenieros con un entrenamiento mínimo para alcanzar resultados prontamente. Por lo que a continuación se mencionan algunos de los paquetes de software disponibles:

- Improver
- Ideator
- Eliminator
- Innovation Workbench ^{TM} (IWB)

3 Trabajo Previo

Desde que TRIZ puede ayudar a los ingenieros y desarrolladores a resolver contradicciones técnicas y a inventar nuevas tecnologías, esto es usado en el desarrollo de nuevos productos muy importantes combinado con otras herramientas, tal como se muestra a continuación:

3.1 TRIZ y Pedagogía

TRIZ pedagogía fue creado con el propósito de enseñar TRIZ. A principios de 1980, intentaron aplicar métodos de TRIZ para enseñar temas académicos, primero con física y química por 12 años. Esto resultó muy efectivo, estudiantes fueron muy exitosos en aprender el material de los cursos. La experiencia entonces se ha ido extendiendo por muchos otros especialistas, quienes han aplica TRIZ para enseñar diversos temas para grupos de diferentes edades, desde pre - escolar a nivel universidad [19].

Para la educación de niños, TRIZ pedagogía tiene un conjunto adicional de objetivos:

- La conservación y reforzamiento de la creatividad natural de los niños.
- La formación de una creatividad con una orientación hacia alcanzar grandes metas.
- La formación de una manera creativa de pensamiento.
- Dominado las técnicas de TRIZ de aprendizaje rápido.

3.2 TRIZ en Mejora de Procesos

TRIZ podría ser usado no solo en innovación de productos, sino también en mejora de procesos. Esto puede incrementar la eficiencia y efectividad en la administración de una organización, lo cual es también valioso. En [26] se proporciona un caso de aplicación de TRIZ en el mejoramientos del proceso de entrenamiento de personal en una organización, TRIZ puede trabajar bien aquí, lo que se propone es aplicar el método Function Analysis para conseguir el problema principal en el proceso de entrenamiento y luego buscar una solución en los principios y leyes de TRIZ.

3.3 TRIZ y QFD

QFD permite identificar las necesidades de los clientes, capacidades tecnológicas organizacionales, propiedades de diseño, problemas de confiabilidad, capacidades funcionales y la relación entre todos estos factores. Una vez que los usuarios conocen las relaciones entre las necesidades del cliente y sus propias habilidades para satisfacer estas necesidades, ellos ayudan a crear productos y diseños innovadores que reúnan o excedan necesidades del cliente. TRIZ es la familia de herramientas que ayuda a los usuarios de QFD a satisfacer sus necesidades de innovar y a sus clientes.

El uso de TRIZ para mejorar la práctica de QFD es Nuevo. Investigaciones continuaran para encontrar formas de integrar estos métodos con la finalidad de ayudar a todos los desarrolladores de proceso y productos a crear soluciones innovadoras que ganen el liderazgo en el mercado resolviendo problemas del cliente. Cada una de las herramientas principales de TRIZ pueden ser usadas en diferentes etapas de QFD. [20, 21, 22, 23 y 24]

3.4 TRIZ y VE

Resolver problemas difíciles es una actividad compleja que es gobernada por la búsqueda de conocimiento. La solución de problemas es afectada por una combinación de procesos de búsqueda y por la disponibilidad del conocimiento requerido para resolver problemas. Esta propuesta describe la aplicación de herramientas basada en conocimiento y analíticas que puedan ser integradas dentro de Value Engineering para mejorar la generación estructurada y utilización de ideas. Las bases para este trabajo es TRIZ el cual está intentando mejorar la efectividad del proceso de resolver problemas y el desarrollo de conceptos de solución implementables.

TRIZ proporciona un enfoque sistemático, estructurado que puede ser usado para aumentar o reemplazar el método de lluvia de ideas tradicional de cómo es aplicado actualmente en

Value Engineering. La utilización de cuestionarios estructurados y de un proceso de formulación de problemas basado en software que revela un exhaustivo conjunto de direcciones para la innovación, mejoraría el esfuerzo creativo de las sesiones de lluvia de ideas a fin de permitir considerar todas las posibles direcciones. Mayores beneficios pueden ser realizados por utilización los resultados de investigaciones y su estructuración en la forma de una base de datos de conocimiento extraído de las patentes a nivel mundial. [18]

3.5 TRIZ y DFSS

Muestra una metodología I-DFSS (Innovative Design For Six Sigma) que combina el pensamiento innovador con herramientas analíticas usadas para diseño de productos, servicios y procesos en un único proceso. El proceso de diseño y de desarrollo algunas veces trabaja independientemente de otras funciones dentro de una organización, ahora el nuevo enfoque para alcanzar cambios estratégicos combina DFSS (Desgin For Six Sigma) y Ideation/TRIZ (I-TRIZ), esta combinación puede guiar a mejores resultados y rápidos diseños que permita a las organizaciones a alcanzar una extraordinaria productividad.
El proceso puede crear diseños nuevos o modificados capaces de alcanzar altos niveles de performance que con otros métodos utilizados individualmente. Los niveles de calidad se seis sigma que DFSS proporciona es efectivo pero combinado con I-TRIZ las posibilidades son significativas. I-DFSS expande el alcance de oportunidades disponibles para una organización por corresponder tecnología, producto y servicios a clientes y mercado. [16]

3.6 TRIZ para Software

TRIZ ha llegado a ser una poderosa herramienta para resolver problemas y crear soluciones inventivas, ayudando a generar un gran número de patentes.
TRIZ es principalmente usada en problemas mecánicos ó eléctricos y no muchos para problemas de software. El software está llegando a ser la fuente de muchos problemas entonces TRIZ se expandió para evitar estos problemas, lo cual ha sido exitoso pero de acuerdo a muchos autores aún existen muchos trabajos necesarios por hacer [39, 41, 42 y 43].

3.6.1 Pilares

Explorando los pilares de TRIZ, se encontró una versión de negocios de TRIZ [34 y 44] lo cual sugiere que existe una conexión de TRIZ con otras filosofías, por lo que se exploran estos pilares en el contexto de software, tal como se muestra en la **Figura** 3. .

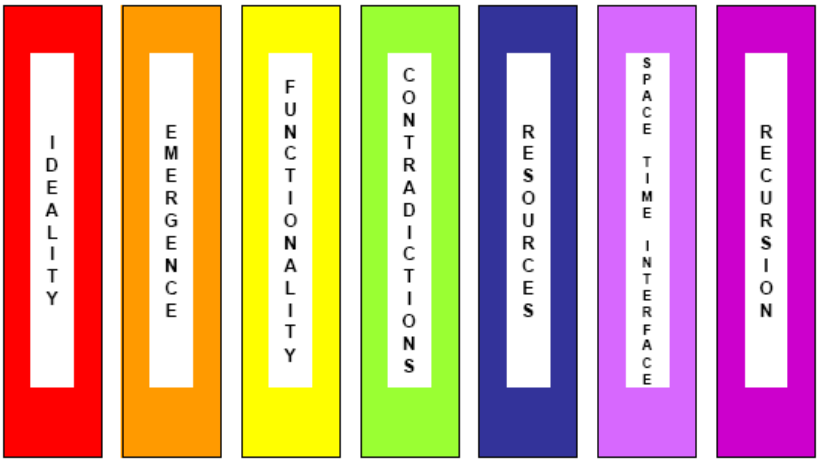

Figura 3.1. Siete Pilares de TRIZ para Software. [2]

Idealidad.- todas las innovaciones exitosas involucran en una dirección el incremento de idealidad, obteniendo más beneficios, menos costo y menos daños. La evolución hacia un resultado final ideal ocurre a través de una serie de patrones que son repetidos a través de diferentes industrias. Absolutamente importante para el diseño y evolución de software.

Surgimiento.- un pilar introducido sobre en la versión de negocios de TRIZ, ya que los problemas de negocios involucran sistemas complejos así cualquier estrategia para generar soluciones apropiadas debe considerar correctamente asuntos de complejidad. Aunque no todos los sistemas de software son complejos, sus interfases con el mundo exterior lo están incrementando. El incremento proviene de la lógica difusa, algoritmos genéticos y sistemas de software basados en agentes así forman sistemas complejos que no pueden ser ignorados por cualquier método que resuelva problemas de software.
Funcionalidad.- los clientes principalmente compran funciones (beneficios), sin embargo el productor debería enfocarse sobre las funciones liberadas en sus productos y servicios. Si los clientes encuentran una mejor manera de alcanzar una función ellos pararían de comprar nuestros productos o servicios. También es importante en el contexto de software ya que el software existe para liberar una función útil ya sean piezas de hardware, software o humanos.
Contradicción.- los sistemas se desarrollan en la dirección de incrementar la idealidad a través del surgimiento continuo y solución de conflictos o contradicciones. Nosotros podemos observar la presencia de contradicciones y su eliminación como el principal conductor de la evolución del software. Las contradicciones tienden a ocurrir en las interfaces entre el software y el mundo exterior más “dentro” del software.
Recursos.- nada dentro o alrededor de un sistema que no se está siendo usado para su máximo potencial es un recurso. Parece ser importante en el contexto de la relación entre el software y el mundo exterior, pero menos importante cuando pensamos acerca del software en sí y la importancia del uso efectivo de los recursos, sin considerar que aún se continúa aplicando la Ley de Moore. Nosotros sin embargo notamos que los programadores utilizan considerables cantidades innecesarias de espacio de memoria cuando construyen una pieza de software simplemente porque la memoria es barata y es fácil escribir un gran algoritmo que uno compacto y elegante.
Espacio/Tiempo/Interfase.- la mente humana esta sujeta a un efecto conocido como inercia psicológica no es adecuado buscar situaciones de un solo ángulo. Cuando estamos tratando de mejorar un sistema, necesitamos ser capaces de cambiar nuestra perspectiva de esto, estos cambios de perspectiva pueden involucrar espacio físico ó virtual, la manera en la cual diferentes elementos de un sistema se relaciona con otros. Este pilar es uno de los más importantes en diseño de software como en el mundo físico, porque hay muchos ingenieros de software que operan en un mundo virtual divorciado de las interfaces con los usuarios actuales. Entonces herramientas que ayuden a los ingenieros de software a ver su mundo de diferentes perspectivas son considerados extremadamente importantes.
Recursión.- este pilar surge de las investigaciones cibernéticas del Dr. Stafford Beer quien dice que existen ciertos fenómenos que se repiten en diferentes niveles jeráquicos (recursividad).

3.6.2 Resolviendo Contradicciones de Software

La clásica matriz de contradicción de TRIZ ha sido el tema de muchas actualizaciones, considerando la primera publicación la “Matriz 2003” [13] y la más reciente una matriz para problemas de negocios [12] . Asimismo, se consideraron los problemas mostrados por los ingenieros de software y encontrando muchas patentes que han tenido algo que contribuir a la construcción de la herramienta.
En el 2003 se decidió que era necesario dar inicio a la creación de la matriz para problemas de software por lo que actualmente Darrell Mann está trabajando en el libro “Triz for Software Engineers”, quien a identificado los parámetros que forman parte de esta matriz (Ver Figura 3.2).

Size (Static)
Size (Dynamic)
Amount of Data
Interface
Speed
Accuracy
Stability
Ability to Detect/Measure
Loss of Time
Loss of Data
Harmful Effects Generated By System
Adaptability/Versatility
Compatibility/Connectability
Ease Of Use
Reliability/Robustness
Security
Aesthetics/Appearance
Harmful Effects On System
System Complexity
Control Complexity
Automation

Figura 3.2. Los 21 Parámetros de la nueva Matriz de Software. [2]

El método de operación de la nueva matriz es exactamente la misma que está siendo usada en otras matrices, el usuario tiene primero que identificar lo que ellos desean mejorar en su sistema, entonces que es lo está impidiendo que mejore. Esto es entonces necesario trasladar las especificaciones en conflicto en un par de la lista de parámetros de la matriz o los que se acerquen más a las especificadas. De aquí la matriz revelaría los principios inventivos principales usados por otros para exitosamente tratar con el conflicto.
La nueva matriz resulta del estudio de alrededor 40,000 patentes de software y soluciones de diseño. El número es limitado por el hecho que actualmente solo USA está concediendo patentes de software (muchos de los cuales después de examinarlos parece que tienen bajo grado de novedad) y entonces los métodos usados en diseños no patentados son usualmente ocultos.

3.6.3 Principios inventivos para software

De acuerdo a las investigaciones realizadas por Darrell Mann[2] se confirmó la existencia de los 40 principios en las que se basa TRIZ clásico sólo se tuvo que adaptar algunas descripciones de los principios al contexto de software, pero la mayoría de ellos se han mantenido, la lista de los principios son mostrados en la **Figura 3.3** (donde los títulos resaltados de verde cambian relativamente de la lista original).
Las investigaciones de TRIZ para software han generado una descripción comprensiva de cada uno de los principios junto con un grupo de ejemplos para cada uno de ellos. La principal meta de tal lista como con la lista de Rea [41 y 42] y otras listas de principios de otros dominios, es proporcionar una lista de ejemplos para explicar el significado de cada uno de los principios, en la esperanza que los usuarios puedan empezar a entenderlos y ser capaces de conectar esto con el problema que ellos están trabajando.

Figura 3.3. Los 40 Principios Inventivos de Software. [2]

3.7 Proceso para el desarrollo de Arquitecturas de Software basado en DFSS

Actualmente en la Maestría en Ingeniería de Software se está desarrollando un proceso para desarrollar una arquitectura de software llamado PASWDFSS [27], el cual se encuentra basado en el proceso DFSS y las diferentes aportaciones que fueron descritas en la sección anterior.

PASWDFSS es un proceso para desarrollar arquitecturas de software de tal manera que permite crear una nueva arquitectura a partir de requerimientos arquitectónicos. Los pasos que contiene el proceso son:

1. Identificación de requerimientos de la Arquitectura
2. Caracterización del diseño de la Arquitectura
3. Documentación de la Arquitectura
4. Optimizar el diseño de la Arquitectura
5. Validación del diseño de la Arquitectura

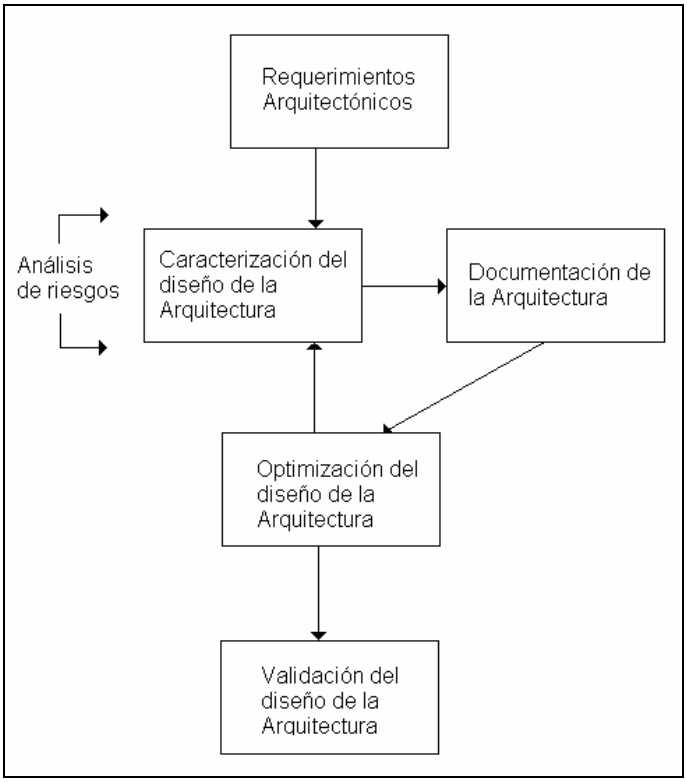

Figura 3.4. Proceso de desarrollo de Arquitecturas de Software basado en DFSS [27]

En la **Figura 3.4**Figura , se muestra el modelo general de la solución propuesta para el proceso de desarrollo de arquitecturas de software. Donde puede observarse:

- Etapa de requerimientos.- la cual tiene por objetivos principales: la creación del equipo de trabajo, selección de los métodos que se utilizarán para la elicitación de requerimientos, la obtención de los mismos, la traducción de los requerimientos a requerimientos funcionales y atributos de calidad medibles y finalmente la categorización de los requerimientos.
- Etapa de Caracterización del diseño de la Arquitectura.- el objetivo principal es la creación de alterativas de diseño, es decir la creación de entidades de diseño y sus relaciones con el objetivo de que cumplan los requerimientos funcionales y atributos de calidad. Dentro de esta fase se encuentra inmerso uno de los pasos de gran importancia: la evaluación de la arquitectura, que como sabemos el principal objetivo de esta etapa es verificar que la arquitectura de software propuesta este cumpliendo con los requerimientos funcionales y de calidad o bien si existe un conflicto entre ellos que este impidiendo la combinación de los mismos.
- Etapa de documentación.- tiene como objetivo el representar la arquitectura en prosa y usando una notación estándar y una plantilla según ANSI/IEEE 1471-2000.
- Etapa de optimización del diseño de la Arquitectura.- el principal objetivo en base a la evaluación del diseño generado, poder hacer los reajustes necesarios para poder contemplar en mayor medida todos los atributos de calidad.
- Etapa de validación.- cuyo objetivo es validar que la arquitectura se encuentre de acuerdo con los requerimientos establecidos por los stakeholders.

4 Descripción del problema

Como se mencionó en secciones anteriores, la competencia entre tecnologías y servicios esta aumentando cada vez mas y estamos una era de crisis y solamente quienes puedan resolver problemas creando nuevas ideas e implementándolas rápidamente podrán sobrevivir. La habilidad de crear y la capacidad de resolver problemas para individuos y organizaciones, son capacidades fundamentalmente basadas en la mente humana. Entonces cómo podríamos incrementar nuestra creatividad?. Recientemente, TRIZ ha destacado por tener la posibilidad de cambiar dramáticamente esta situación.

Un problema, dice Edward de Bono, es simplemente *"la diferencia entre lo que nosotros tenemos y lo que deseamos"*. Altshuller creyó que los problemas surgen de contradicciones o trade-off entre dos o más elementos, tal como:"si necesitamos mas aceleración, necesitamos una máquina grande pero esto incrementaría el costo de un carro", esto significa que algo deseable también trae algo no deseable ó menos de algo deseable. Por lo que una situación inventiva puede involucrar varias contradicciones y para resolverlas el inventor necesita desarrollar enfoques creativos para disolver la contradicción, entonces debería inventar una máquina que produzca más aceleración sin incrementar su costo.

Entonces observamos que en diferentes áreas se presentan problemas como contradicciones, en especial en el área de software identificamos problemas relacionados con los aspectos críticos en el diseño y en a construcción de la arquitectura de un sistema de software. Actualmente el proceso de diseño de la arquitectura de software no esta del todo formalizado y a menudo es más intuitivo que racional. Algunos de los problemas más críticos a los que se han enfrentado los arquitectos de software y que han sido tema de estudio de los investigadores son:

- La actividad más compleja durante el desarrollo de aplicaciones es la transformación de la especificación de los requerimientos en la implementación del sistema [23].
- Los requerimientos de calidad influyen fuertemente a la arquitectura de software y según la experiencia de éstos son generalmente tratados con un proceso informal durante el diseño de la arquitectura. Siendo el principal foco de atención los requerimientos funcionales del sistema, por lo que no se presta mucha atención en los requerimientos de calidad [24].
- Los procesos de arquitecturas de software no están definidos de una manera detallada y no se encuentran en una mejora continua.

A pesar de los progresos que ha habido en el área de arquitectura de software, este campo permanece todavía inmaduro. Por lo que el presente trabajo tiene como objetivo integrar TRIZ sobre el proceso de desarrollo de arquitecturas de software basado en DFSS para reforzar el proceso de diseño de una arquitectura, asegurar que nuevos productos son diseñados reuniendo altos niveles de calidad, para resolver contradicciones en los requerimientos de diseño, acortar el tiempo de desarrollo de una nueva arquitectura, reducir errores, disminuir costos, mejorar la satisfacción del cliente produciendo productos amigables con el usuario. Además, de permitir a los estudiantes, Ingenieros de Software y Organizaciones a estimular su pensamiento innovador para resolver problemas difíciles relacionados al diseño de una nueva arquitectura.

5 Propuesta de Solución

La propuesta del presente trabajo, basada en las investigaciones mostradas en la **sección 3,** es la integración de TRIZ sobre el proceso de desarrollo de arquitecturas de software basada en DFSS.

En el desarrollo de software, la fase de arquitectura es un tiempo durante el cual el software está mapeado a través del uso de diagramas de diseño y prototipos. Típicamente la arquitectura de un sistema también es conocida como diseño ó diseño de alto nivel, la cual es descrita en un documento llamado Documento de Arquitectura de Software – SAD. Como en una construcción, la fase de la arquitectura proporciona un significado de exploración de alternativas a un bajo costo relativo, para software una arquitectura proporciona una estructura técnica para un proyecto, por lo que una buena arquitectura hace que el resto del trabajo sea fácil, en cambio una mala arquitectura hace que el resto del trabajo sea casi imposible [27].

En la creación de arquitecturas uno tiene que tratar con conflictivas demandas, es decir la arquitectura tiene que cumplir con requerimientos funcionales y no funcionales. Ejemplo de requerimientos no funcionales son: portabilidad, mantenibilidad, flexibilidad, extendibilidad y usabilidad. En algunas áreas de aplicación una gráfica de interdependencia de “softgoal” son usadas para visualizar conflictos y las mejores soluciones son las adecuadas entre esas demandas conflictivas, difícilmente todas las demandas son totalmente encontradas.

Pensando acerca de la integración de herramientas ó métodos de TRIZ con el proceso de desarrollo de arquitecturas de software (Ver Sección **3.7**) se lograría una nueva metodología estructurada para mejorar la calidad inventiva llamada I- PASWDFSS (I: Innovación y PASWDFSS: Proceso de Arquitectura de Software basado en DFSS). Por lo que se han identificado 3 maneras que las herramientas o métodos de TRIZ pueden integrarse en el proceso de desarrollo de software, tal como se muestra en la **Figura 5.1:**

- La primera integración de herramientas de TRIZ tales como: análisis Preliminar, ISQ, ARIZ y IFR pueden ayudar a identificar los requerimientos de la arquitectura y definir correctamente el problema sobre el cual puede ser aplicado (Etapa de Requerimientos). Además las herramientas de Su – Field model, los 76 Estándares y Patrones de Evolución pueden ayudar en la creación o mejora de la medición del sistema.
- La segunda integración de herramientas de TRIZ tales como Matriz de Contradicción, los 40 Principios Innovadores y Principios de Separación puede ayudar en la definición y solución de contradicciones entre variables o requerimientos con la finalidad de realizar un adecuado diseño (Etapa de Caracterización).
- La tercera integración de herramientas de TRIZ tales como: Leyes de Evolución de la Idealidad y AFD pueden ayudar en cada una de las etapas del proceso de desarrollo de una arquitectura dirigida a la idealidad e Innovación. Además de que AFD puede ser usado para predecir las formas de fallas y resolver la causa raíz lo cual ayudaría a mejorar la calidad del diseño de la arquitectura (Etapa de Optimización).

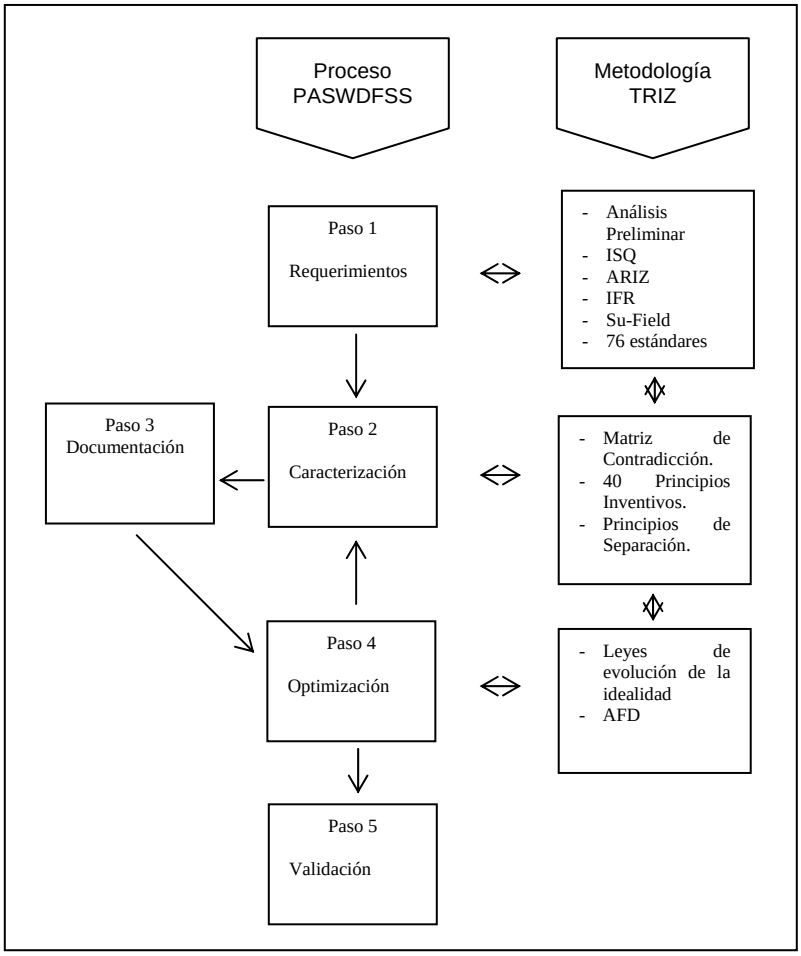

Figura 5.1. Integración de la metodología TRIZ sobre el Proceso PASWDFSS

Por lo que combinando TRIZ durante el proceso de desarrollo de arquitecturas podría ser muy útil en las diferentes etapas del proceso, principalmente en la etapa de diseño permitiendo dar solución a demandas conflictivas (contradicciones) de una manera satisfactoria. Desde que las arquitecturas son también utilizadas en otros campos como la ingeniería mecánica podemos aprender de la aplicación de TRIZ en ese campo. La creatividad e innovación son esenciales en todas partes por lo que TRIZ ya no sería solo para ingenieros y diseñadores, ya que durante un proyecto se trata con equipos de ingenieros que deben resolver problemas reales de software y los beneficios de la aplicación de TRIZ serían:

- Productos de alta calidad.
- Altas utilidades.
- Productos innovativos.
- Anticipando fallas futuras.
- Protección del capital intelectual.
- Inventar siguientes generaciones.

6 Conclusiones y Trabajo Futuro

Entre las conclusiones se pueden mencionar:

- TRIZ acumula y concentra todo el conocimiento humano respectivo y entonces lo aplica para resolver problemas de inventiva. La primera meta de TRIZ es transformar actuales problemas de inventiva a problemas de rutina en el futuro. La segunda meta es desarrollar personas con inventiva, capaces de resolver problemas no técnicos y técnicos creativamente.

- Debido al gran desarrollo por parte de grupos de investigadores, TRIZ ha llegado a convertirse en una metodología poderosa para el pensamiento inventivo y efectivo durante la solución de problemas técnicos no rutinarios. Esto ayuda a cumplir el axioma principal de nuestra sociedad:
  "La cantidad y calidad de los requerimientos humanos y necesidades incrementan con el tiempo".
  Esto sucede porque TRIZ hace posible satisfacer requerimientos humanos para resolver problemas técnicos y no técnicos en un corto tiempo y con alta calidad.

- Como cualquier otra ciencia TRIZ puede ser desarrollada en una de dos direcciones:
  - Por una manera empírica de observación, descripción, abstracción, generalización, formulación de hipótesis y guías ó
  - Asumiendo un conjunto de axiomas y postulados, proporcionado teoremas, modelando, refinando y formulando una teoría.

  Ahora la primera dirección domina debido a la gran influencia de Altshuller y la actividad de sus estudiantes (S.S. Litvin, V.M. Gerasimov, Yu. P. Salamatov, A.I. Gasanov, V.A. Mikhajlow, entre otros) ya que la segunda dirección fue desarrollado por otros (S.D. Savransky, A.M.Pinyaev, G.A. Yezersky, entre otros). Sin embargo, estos enfoques no son mutuamente excluyentes y un sistema factible puede ser obtenido de la investigación en ambas direcciones.

- TRIZ trasmite el origen de los inventos de eventos ocasionales sobre prácticas regulares a través de su habilidad única de ayudar a obtener fuertes soluciones de los siguientes tipos de problemas no rutinarios:
  - Alcanzar un mejoramiento específico, mejorar la calidad y reducir el costo de una técnica existente.
  - Pronosticar una técnica específica y crear la conceptualización de una nueva técnica.

- TRIZ hard (enfoque técnico) y soft (enfoque no técnico) es un intento para proporcionar una metodología general y científicamente comprensivo para la creación de nuevas ideas, conceptos y soluciones innovadoras. Esta es una metodología revolucionaria que debería ser desarrollada como una ciencia de la ingeniería de innovación.

- Esta metodología I- PASWDFSS combina el pensamiento innovativo con el proceso de diseño de una arquitectura para reducir el tiempo de desarrollo y asegurando un producto de mejor calidad. Esta combinación puede guiar a un mejor y más rápido diseño que permitirá tanto a estudiantes u organizaciones alcanzar una productividad extraordinaria.

- Como lo menciona Karasik [14] se cree que mientras los principios de TRIZ clásico se apliquen, ellos formaran una parte insuficiente pero necesaria de todo el almacén de innovación sistemática para software. Por lo menos, los hallazgos de las investigaciones

de TRIZ para software han revelado la necesidad de incorporar ideas de teoría de complejidad y cibernética sobre la filosofía básica, esto es probablemente aun los primeros días en la historia de la evolución general. Sin embargo, creemos que es posible crear herramientas capaces de ayudar a los ingenieros de software a hacer mejor su trabajo. Específicamente, han surgido metodologías de definición y solución tales como: Resultado Final Ideal, Matriz de Contradicción, Principios inventivos, Self-X, entre otros.

Entre los trabajos a futuro se pueden identificar:

- Se pueden identificar cuatro corrientes de aplicación posibles para TRIZ en los cuales se pueden obtener trabajos futuros:
  - **TRIZ teórico**.- creación de nuevas ideas y conceptos , búsqueda y reconocimiento de nuevas heurísticas, construcción de nuevos instrumentos, representación e integración del conocimiento de otras ciencias para resolver problemas, tanto como el análisis, desarrollo y descripción matemática de conocidas ideas y conceptos de TRIZ .
  - **TRIZ experimental**.- probar los nuevos instrumentos y heurísticas durante la solución de problemas reales y diseño de técnicas, aplicación de heurísticas e instrumentos de TRIZ conocidas en nuevos campos de la ingeniería donde estos no han sido aplicados.
  - **TRIZ aplicada**.- uso de instrumentos y heurísticas de TRIZ conocidas para resolver problemas particulares en varios campos de la ingeniería donde estos ya han trabajado.
  - **TRIZ pedagógico**.- desarrollo de instrumentos y métodos para la enseñanza efectiva de TRIZ, búsqueda de soluciones conocidas de problemas de fácil entendimiento y su presentación en la educación, tanto como en la implementación de TRIZ en la enseñanza de otras ciencias, arte liberal y metodologías.

- Desarrollos futuros de TRIZ pueden ser categorizados de acuerdo a los siguientes tres atributos:
  - Área de validez (general, adecuado a cualquier técnica para un área en particular de la ingeniería).
  - Enfoque (conclusiones teóricas obtenidas por inducción vs estudios experimentales de todas las patentes encontradas y otras fuentes de información).
  - Contenido (resultados teóricos vs casos de estudio aplicados).

- Las siguientes áreas son donde aplicaciones de TRIZ pueden ser desarrolladas:
  - Un principio básico para una teoría de resolver problemas no rutinarios.
  - Una parte principal para el conocimiento coordinado y extendido sobre el origen de técnicas.
  - Un punto de inicio y concepto coordinado para teorías especiales de sistemas técnicos y proceso tecnológicos.
  - Una fuente de conocimiento para la ingeniería del diseño.
  - Una base para las ciencias psicológicas tratando con creatividad.
  - Una esencia para los procesos y procedimientos de la ingeniería inventiva.
  - Un punto de inicio para nuevos software: base de datos, algoritmos, inteligencia artificial, sistemas basados en conocimiento, entre otros.
  - Una base para un enfoque educacional integrado para ingenieros.
  - Una base para la comunicación y entendimiento entre ingenieros de diferentes campos, administradores y personas no técnicas.

7 Glosario de términos

Término	Descripción
Creatividad	Es un proceso que involucra la generación de nuevas ideas o conceptos o nuecas asociaciones entre ideas y conceptos existentes. Los productos de creatividad pueden usualmente tener originalidad y aplicabilidad.
Patente	Es la certificación que el gobierno otorga , tanto a personas físicas como morales, la cual les permite explotar exclusivamente invenciones que consistan en nuevos productos o procesos durante un plazo improrrogable de 20 años contados a partir de la presentación de la solicitud correspondiente.
Ley de la Propiedad Industrial	Establece que serán patentables las invenciones que sean nuevas, resultado de una actividad inventiva y susceptible de aplicación industrial.
Idealidad	Es una abstracción que representa las reflexiones de la realidad útil para los estudios de varios fenómenos.
Heurística	Son criterios, métodos o principios para decidir cual entre diferentes alternativas de una acción prometida será la más efectiva para alcanzar alguna meta. Además, puede ser una “rule of thumb” ó principio de amplia aplicación que es usado para guiar acciones.
Diseño	Proceso de conceptualizar, inventar o idear un esquema para cambiar una especificación para un programa de computación en un programa operacional. Esta actividad relaciona el desarrollo de requerimientos a su construcción.
Psychological Inertial	Es el esfuerzo hecho por un sistema para preservar un estado estable o resistir a un cambio de estado, se enfoca en la mente sobre lo que es conocido. Esta inercia solo es buena cuando la dirección de la solución es reconocida correctamente.
Arquitectura	Especificación de alto nivel de la manera en la cual el problema sería resuelto. Esto es un plan para construir la solución correcta al problema.
Trade-off	Cuando alternativas son analizadas y se tiene que aceptar menos de una cosa para obtener mas de otra.
Critical to Quality	Son características medibles claves de un producto o proceso cuyos estándares de performance ó límites de especificaciones que deben ser encontrados para satisfacer al cliente.

8 Acrónimos

Término	Descripción
TRIZ	Acrónimo ruso: Teoriya Resheniya Izobretatelskikh Zadatch
TIPS	Siglas en inglés: Theory of Inventive Problems Solving.
QFD	Quality Function Deployment
ISQ	Innovation Situation Questionnaire
IFR	Ideal Final Result
AFD	Anticipatory Failure Determination
ARIZ	Algorithm for Inventive Problem Solving
CTQ	Critical To Quality
FA	Failure Analysis

9 Referencias

[1] CSemyon D. Savransky. "Engineering of Creativity; Introduction to TRIZ Metodology of Inventive Problem Solving" 2000.

[2] Mann, D.L., TRIZ For Software. The TRIZ Journal. Oct. 2004. Internet: http://www.triz-journal.com/archives/2004/10/04.pdf

[3] Fulbright, R., TRIZ and Software Fini. The TRIZ Journal. Aug. 2004. Internet: http://www.triz-journal.com/archives/2004/08/02.pdf

[4] Rea, K.C., TRIZ and Software 40 Principles Analogies, Part 2. The TRIZ Journal. Nov,2001. Internet: http://www.triz-journal.com/archives/2001/11/e/

[5] Rea, K.C., Applying TRIZ to Software Problems - Creatively Bridging Academia and Practice in Computing . TRIZCON 2002 and The TRIZ Journal. Oct., 2002. Internet: http://www.triz-journal.com/archives/2002/10/c/index.htm

[6] Nakagawa, Toru, Software Engineering and TRIZ (1) Structured Programming Reviewed with TRIZ. To appear in TRIZCON 2005 - April 2005. Internet: http://www.aitriz.org/

[7] Rea, K.C. Applying TRIZ to Software Problems. TRIZCON2002.

[8] Michael Schlueter . “Fast Software by TRIZ”. ETRIA World Conference – TRIZ Future 2003.

[9] Rea, K.C.TRIZ and Software – 40 Principle Analogies, Part 1. TRIZ-journal 2001.

[10] Rea, K.C.TRIZ and Software – 40 Principle Analogies, Part 2. TRIZ-journal 2001.

[11] Herman Hartmann, Ad Vermeulen and Martine van Beers. Application of TRIZ in Software Development. TRIZ-journal 2004.

[12] Mann, D.L., ‘Hands-On Systematic Innovation For Business & Management’, IFR Press, August 2004.

[13] Mann, D.L., Dewulf, S., Zlotin, B., Zusman, A., ‘Matrix 2003: Updating The TRIZ Contradiction Matrix’, CREAX Press, Belgium, July 2003.

[14] Karasik, Y., editorial comment, Anti-TRIZ Journal, Vol.2, No.2, September 2004.

[15] Steve McConnel. Book: Software Project Survival Guide.1998.

[16] Joseph A. De Feo y Zion Bar-El. Creating strategic change more efficiently with a new Design for Six Sigma Process. Journal of Change Management. August 2002.

[17] Ellen Domb, The PQR Group, Upland, CA USA. Managing Creativity for Project Success. TRIZ-journal 2000.

[18] Dana W. Clarke. Integrating TRIZ with Value Engineering: Discovering Alternatives to Traditional Brainstorming and the Selection and Use of Ideas. International SAVE Conference, San Antonio, Texas. 1999.

[19] Boris Zlotin and Alla Zusman. TRIZ and Pedagogy. Ideation International. July 1991. http://www.ideationtriz.com/

[20] Walter P. Bond, Turky N. Al-Otaiby, Mohsen N. AlSharif. Software Architecture Assessment Using Quality Function Deployment. Department of Computer Sciences-Florida Institute of Technology. 2002.

[21] Michael Schlueter.QFD by TRIZ. TRIZCON2001. March, 2001

[22] John Terninko. The QFD, TRIZ and Taguchi Connection: Customer-Driven Robust Innovation. The Ninth Symposium on Quality Function Deployment. June 1997.

[23] Noel León-Rovira and HUmberto Aguayo.A New Model of the Conceptual Design Process Using QFD/FA/TRIZ. Instituto Tecnológico y de Estudios Superiores de Monterrey.1997.

[24] Garlan David. Software Architecture: a Roadmap. School of Computer Science Carnegie Mellon University Pittsburgh, PA.

[25] Bosh Jan, Molin Peter. Software Architecture Design: Evaluation and Transformation. University of Karlskrona/Ronneby, Departement of Computer Science.

[26] Haiyan Ru y Haibo Ru. Applying TRIZ in Process Improvement

[27] CIMAT- Maestría en Ingeniería de Software: Nuñez, Araceli. Proceso para el desarrollo de Arquitecturas de Software basado en DFSS. Agosto 2006 (Investigación en Proceso).

Apéndice A. Principios Inventivos para Software

1. Segmentation a. Dividing an object into independent parts. b. Make an object modular. c. Increase the degree of fragmentation.	Intelligent Agents C++ templates Confidential Objects
2. Extraction Separate interfering or necessary parts	Extraction of text in images
3. Local Quality a. Change structure from uniform to non-uniform b. Make parts perform different functions	Non-uniform access algorithms Higher levels in a single index tree
4. Asymmetry Change from symmetrical to asymmetrical.	Load balancing, resource allocation
5. Consolidation Make operations contiguous or parallel	Threading, multitasking
6. Universality Perform multiple functions; eliminate parts	Personalization of user interface
7. Nesting Place an object into another	Classes within other classes
8. Counterweight Counter weight with lift	Shared objects in multiple contexts
9. Prior counteraction Preload compensating counter tension	Pre-processing
10. Prior action Fully or partially act before necessary	Pre-compiling
11. Cushion in advance Prepare beforehand to compensate low reliability	Fair scheduling in packet networks
12. Equipotentiality In a potential field, limit position changes	A transparent persistent object store
13. Do it in reverse Invert actions	Transaction rollback
14. Spheroidality Replace linear parts with curved parts	Circular abstract data structures
15. Dynamicity Find an optimal operating condition	Dynamic Linked Libraries (DLLs)
16. Partial or excessive action Use “slightly less” or “slightly more”	Perturbation analysis
17. Transition into new dimension Move in more dimensions	Multi-layered assembly of classes
18. Mechanical Vibration Oscillation	Change the rate of an algorithm
19. Periodic Action Periodic or pulsating actions	Scheduling algorithms
20. Continuity of useful action a. Continue actions b. Eliminate all idle or intermittent actions	Utilizing processor at full load Eliminatel blocking processes

Table 1a – Summary of TRIZ Analogs for Software (1-20)

21. Rushing through Conduct a process at high speed	Burst-mode transmission
22. Convert harm into benefit Eliminate the primary harmful action	Bottleneck DDOS zombies
23. Feedback Introduce feedback	Feedback improving iterations
24. Mediator Use an intermediary	XML-based view generation
25. Self-service Performing auxiliary functions	Periodic auto-update
26. Copying Use simpler and inexpensive copies	Perform a shallow copy
27. Dispose Use multiple inexpensive objects	Rapid prototyping
28. Replacement of Mechanical System Replace mechanical means	Voice recognition/dictation
29. Pneumatic or hydraulic construction Use inflatable gas or liquid parts	Dynamically allocated data structures
30. Flexible films or thin membranes Isolate the object from the environment	Wrapper objects
31. Porous materials Make an object porous	Intelligent tutoring systems
32. Changing the color Change the external view (transparency)	Transparency layers
33. Homogeneity Use same material	Container objects
34. Rejecting and regenerating parts a. Discard portions of an object b. Restore consumable parts	Garbage collection Transaction rollback
35. Transformation properties Change the degree or flexibility	Multi-role objects
36. Phase transition Phase transition phenomenon	Emergent behavior
37. Thermal expansion Use thermal expansion or contraction	System memory
38. Accelerated oxidation Use oxygen-enriched air	Salted encryption
39. Inert Environment Replace normal environment with an inert one	Test harness
40. Composite materials Use composite (multiple) materials	Composite objects

Table 1b – Summary of TRIZ Analogs for Software (21-40)

Apéndice B. Los 39 parámetros de la Matriz de Contradicción de Altshuller

No.	Title	Explanation
	Moving objects	Objects which can easily change position in space, either on their own, or as a result of external forces. Vehicles and objects designed to be portable are the basic members of this class.
	Stationary objects.	Objects which do not change position in space, either on their own, or as a result of external forces. Consider the conditions under which the object is being used.
1	Weight of moving object	The mass of the object, in a gravitational field. The force that the body exerts on its support or suspension.
2	Weight of stationary object	The mass of the object, in a gravitational field. The force that the body exerts on its support or suspension, or on the surface on which it rests.
3	Length of moving object	Any one linear dimension, not necessarily the longest, is considered a length.
4	Length of stationary object	Same.
5	Area of moving object	A geometrical characteristic described by the part of a plane enclosed by a line. The part of a surface occupied by the object. OR the square measure of the surface, either internal or external, of an object.
6	Area of stationary object	Same
7	Volume of moving object	The cubic measure of space occupied by the object. Length x width x height for a rectangular object, height x area for a cylinder, etc.
8	Volume of stationary object	Same
9	Speed	The velocity of an object; the rate of a process or action in time.
10	Force	Force measures the interaction between systems. In Newtonian physics, force = mass X acceleration. In TRIZ, force is any interaction that is intended to change an object's condition.
11	Stress or pressure	Force per unit area. Also, tension.
12	Shape	The external contours, appearance of a system.
13	Stability of the object's composition	The wholeness or integrity of the system; the relationship of the system's constituent elements. Wear, chemical decomposition, and disassembly are all decreases in stability. Increasing entropy is decreasing stability.
14	Strength	The extent to which the object is able to resist changing in response to force. Resistance to breaking .

15	Duration of action by a moving object	The time that the object can perform the action. Service life. Mean time between failure is a measure of the duration of action. Also, durability.
16	Duration of action by a stationary object	Same.
17	Temperature	The thermal condition of the object or system. Loosely includes other thermal parameters, such as heat capacity, that affect the rate of change of temperature.
18	Illumination intensity * (jargon)	Light flux per unit area, also any other illumination characteristics of the system such as brightness, light quality, etc..
19	Use of energy by moving object	The measure of the object's capacity for doing work. In classical mechanics, Energy is the product of force times distance. This includes the use of energy provided by the super-system (such as electrical energy or heat.) Energy required to do a particular job.
20	Use of energy by stationary object	same
21	Power * (jargon)	The time rate at which work is performed. The rate of use of energy.
22	Loss of Energy	Use of energy that does not contribute to the job being done. See 19. Reducing the loss of energy sometimes requires different techniques from improving the use of energy, which is why this is a separate category.
23	Loss of substance	Partial or complete, permanent or temporary, loss of some of a system's materials, substances, parts, or subsystems.
24	Loss of Information	Partial or complete, permanent or temporary, loss of data or access to data in or by a system. Frequently includes sensory data such as aroma, texture, etc.
25	Loss of Time	Time is the duration of an activity. Improving the loss of time means reducing the time taken for the activity. "Cycle time reduction" is a common term.
26	Quantity of substance/the matter	The number or amount of a system's materials, substances, parts or subsystems which might be changed fully or partially, permanently or temporarily.
27	Reliability	A system's ability to perform its intended functions in predictable ways and conditions.
28	Measurement accuracy	The closeness of the measured value to the actual value of a property of a system. Reducing the error in a measurement increases the accuracy of the measurement.
29	Manufacturing precision	The extent to which the actual characteristics of the system or object match the specified or required characteristics.
30	External harm affects the object	Susceptibility of a system to externally generated (harmful) effects.
31	Object-	A harmful effect is one that reduces the efficiency or quality of the functioning

	generated harmful factors	of the object or system. These harmful effects are generated by the object or system, as part of its operation.
32	Ease of manufacture	The degree of facility, comfort or effortlessness in manufacturing or fabricating the object/system.
33	Ease of operation	Simplicity: The process is NOT easy if it requires a large number of people, large number of steps in the operation, needs special tools, etc. "Hard" processes have low yield and "easy" process have high yield; they are easy to do right.
34	Ease of repair	Quality characteristics such as convenience, comfort, simplicity, and time to repair faults, failures, or defects in a system.
35	Adaptability or versatility	The extent to which a system/object positively responds to external changes. Also, a system that can be used in multiple ways for under a variety of circumstances.
36	Device complexity	The number and diversity of elements and element interrelationships within a system. The user may be an element of the system that increases the complexity. The difficulty of mastering the system is a measure of its complexity.
37	Difficulty of detecting and measuring	Measuring or monitoring systems that are complex, costly, require much time and labor to set up and use, or that have complex relationships between components or components that interfere with each other all demonstrate "difficulty of detecting and measuring." Increasing cost of measuring to a saticfactory error is also a sign of increased difficulty of measuring.
38	Extent of automation	The extent to which a system or object performs its functions without human interface. The lowest level of automation is the use of a manually operated tool. For intermediatel levels, humans program the tool, observe its operation, and interrupt or re-program as needed. For the highest level, the machine senses the operation needed, programs itself, and monitors its own operations.
39	Productivity *	The number of functions or operations performed by a system per unit time. The time for a unit function or operation. The output per unit time, or the cost per unit output.

Apéndice C. Matriz de Contradicción de Altshuller

	Worsening Feature → / Improving Feature ↓	Weight of moving object	Weight of stationary object	Length of moving object	Length of stationary object	Area of moving object	Area of stationary object	Volume of moving object	Volume of stationary object	Speed	Force (Intensity)	Stress or pressure	Shape	Stability of the object's composition	Strength	Duration of action of moving object	Duration of action of stationary object	Temperature	Illumination intensity	Use of energy by moving object	Use of energy by stationary object
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Weight of moving object	+	-	15, 8, 29,34	-	29, 17,	-	29, 2, 40, 28	-	2, 8, 15, 38	8, 10, 18, 37	10, 36,	10, 14,	1, 35, 19, 39	28, 27,	5, 34, 31, 35	-	6, 29, 4, 38	19, 1, 32	35, 12,	-
2	Weight of stationary object	-	+	-	10, 1, 29, 35	-	35, 30,	-	5, 35, 14, 2	-	8, 10, 19, 35	13, 29,	13, 10,	26, 39, 1,	28, 2, 10, 27	-	2, 27, 19, 6	28, 19,	19, 32, 35	-	18, 19,
3	Length of moving object	8, 15, 29, 34	-	+	-	15, 17, 4	-	7, 17, 4, 35	-	13, 4, 8	17, 10, 4	1, 8, 35	1, 8, 10, 29	1, 8, 15, 34	8, 35, 29, 34	19	-	10, 15, 19	32	8, 35, 24	-
4	Length of stationary object		35, 28,	-	+	-	17, 7, 10, 40	-	35, 8, 2,14	-	28, 10	1, 14, 35	13, 14,	39, 37, 35	15, 14,	-	1, 10, 35	3, 35, 38, 18	3, 25	-	
5	Area of moving object	2, 17, 29, 4	-	14, 15,	-	+	-	7, 14, 17, 4		29, 30, 4,	19, 30,	10, 15,	5, 34, 29, 4	11, 2, 13, 39	3, 15, 40, 14	6, 3	-	2, 15, 16	15, 32,	19, 32	-
6	Area of stationary object	-	30, 2, 14, 18	-	26, 7, 9, 39	-	+	-		-	1, 18, 35, 36	10, 15,		2, 38	40	-	2, 10, 19, 30	35, 39, 38		-	
7	Volume of moving object	2, 26, 29, 40	-	1, 7, 4, 35	-	1, 7, 4, 17	-	+	-	29, 4, 38, 34	15, 35,	6, 35, 36, 37	1, 15, 29, 4	28, 10, 1,	9, 14, 15, 7	6, 35, 4	-	34, 39,	2, 13, 10	35	-
8	Volume of stationary object	-	35, 10,	19, 14	35, 8, 2, 14	-		-	+	-	2, 18, 37	24, 35	7, 2, 35	34, 28,	9, 14, 17, 15	-	35, 34, 38	35, 6, 4		-	
9	Speed	2, 28, 13, 38	-	13, 14, 8	-	29, 30, 34	-	7, 29, 34	-	+	13, 28,	6, 18, 38, 40	35, 15,	28, 33, 1,	8, 3, 26, 14	3, 19, 35, 5	-	28, 30,	10, 13, 19	8, 15, 35, 38	-
10	Force (Intensity)	8, 1, 37, 18	18, 13, 1,	17, 19, 9,	28, 10	19, 10, 15	1, 18, 36, 37	15, 9, 12, 37	2, 36, 18, 37	13, 28,	+	18, 21, 11	10, 35,	35, 10, 21	35, 10,	19, 2		35, 10, 21	-	19, 17, 10	1, 16, 36, 37
11	Stress or pressure	10, 36,	13, 29,	35, 10, 36	35, 1, 14, 16	10, 15,	10, 15,	6, 35, 10	35, 24	6, 35, 36	36, 35, 21	+	35, 4, 15, 10	35, 33, 2,	9, 18, 3, 40	19, 3, 27		35, 39,	-	14, 24,	
12	Shape	8, 10, 29, 40	15, 10,	29, 34, 5,	13, 14,	5, 34, 4, 10		14, 4, 15, 22	7, 2, 35	35, 15,	35, 10,	34, 15,	+	33, 1, 18, 4	30, 14,	14, 26, 9,		22, 14,	13, 15, 32	2, 6, 34, 14	
13	Stability of the object's composition	21, 35, 2,	26, 39, 1,	13, 15, 1,	37	2, 11, 13	39	28, 10,	34, 28,	33, 15,	10, 35,	2, 35, 40	22, 1, 18, 4	+	17, 9, 15	13, 27,	39, 3, 35, 23	35, 1, 32	32, 3, 27, 16	13, 19	27, 4, 29, 18
14	Strength	1, 8, 40, 15	40, 26,	1, 15, 8, 35	15, 14,	3, 34, 40, 29	9, 40, 28	10, 15,	9, 14, 17, 15	8, 13, 26, 14	10, 18, 3,	10, 3, 18, 40	10, 30,	13, 17, 35	+	27, 3, 26		30, 10, 40	35, 19	19, 35, 10	35
15	Duration of action of moving object	19, 5, 34, 31	-	2, 19, 9	-	3, 17, 19	-	10, 2, 19, 30	-	3, 35, 5	19, 2, 16	19, 3, 27	14, 26,	13, 3, 35	27, 3, 10	+	-	19, 35, 39	2, 19, 4, 35	28, 6, 35, 18	
16	Duration of action by stationary object	-	6, 27, 19, 16	-	1, 40, 35	-		-	35, 34, 38	-				39, 3, 35, 23		-	+	19, 18,		-	
17	Temperature	36,22, 6, 38	22, 35, 32	15, 19, 9	15, 19, 9	3, 35, 39, 18	35, 38	34, 39,	35, 6, 4	2, 28, 36, 30	35, 10, 3,	35, 39,	14, 22,	1, 35, 32	10, 30,	19, 13, 39	19, 18,	+	32, 30,	19, 15, 3,	
18	Illumination intensity	19, 1, 32	2, 35, 32	19, 32, 16		19, 32, 26		2, 13, 10		10, 13, 19	26, 19, 6		32, 30	32, 3, 27	35, 19	2, 19, 6		32, 35, 19	+	32, 1, 19	32, 35, 1,
19	Use of energy by moving object	12,18, 28,31	-	12, 28	-	15, 19, 25	-	35, 13, 18	-	8, 35, 35	16, 26,	23, 14, 25	12, 2, 29	19, 13,	5, 19, 9, 35	28, 35, 6,	-	19, 24, 3,	2, 15, 19	+	-
20	Use of energy by stationary object	-	19, 9, 6, 27	-		-		-		-	36, 37			27, 4, 29, 18	35				19, 2, 35, 32	-	+
21	Power	8, 36, 38, 31	19, 26,	1, 10, 35, 37		19, 38	17, 32,	35, 6, 38	30, 6, 25	15, 35, 2	26, 2, 36, 35	22, 10, 35	29, 14, 2,	35, 32,	26, 10, 28	19, 35,	16	2, 14, 17, 25	16, 6, 19	16, 6, 19, 37	
22	Loss of Energy	15, 6, 19, 28	19, 6, 18, 9	7, 2, 6, 13	6, 38, 7	15, 26,	17, 7, 30, 18	7, 18, 23	7	16, 35, 38	36, 38			14, 2, 39, 6	26			19, 38, 7	1, 13, 32, 15		
23	Loss of substance	35, 6, 23, 40	35, 6, 22, 32	14, 29,	10, 28,24	35, 2, 10, 31	10, 18,	1, 29, 30, 36	3, 39, 18, 31	10, 13,	14, 15,	3, 36, 37, 10	29, 35, 3,	2, 14, 30, 40	35, 28,	28, 27, 3,	27, 16,	21, 36,	1, 6, 13	35, 18,	28, 27,
24	Loss of Information	10, 24, 35	10, 35, 5	1, 26	26	30, 26	30, 16		2, 22	26, 32						10	10		19		
25	Loss of Time	10, 20,	10, 20,	15, 2, 29	30, 24,	26, 4, 5, 16	10, 35,	2, 5, 34, 10	35, 16,		10, 37,	37, 36,4	4, 10, 34, 17	35, 3, 22, 5	29, 3, 28, 18	20, 10,	28, 20,	35, 29,	1, 19, 26, 17	35, 38,	1
26	Quantity of substance/the matter	35, 6, 18, 31	27, 26,	29, 14,		15, 14, 29	2, 18, 40, 4	15, 20, 29		35, 29,	35, 14, 3	10, 36,	35, 14	15, 2, 17, 40	14, 35,	3, 35, 10, 40	3, 35, 31	3, 17, 39		34, 29,	3, 35, 31
27	Reliability	3, 8, 10, 40	3, 10, 8, 28	15, 9, 14, 4	15, 29,	17, 10,	32, 35,	3, 10, 14, 24	2, 35, 24	21, 35,	8, 28, 10, 3	10, 24,	35, 1, 16, 11		11, 28	2, 35, 3, 25	34, 27, 6,	3, 35, 10	11, 32, 13	21, 11,	36, 23
28	Measurement accuracy	32, 35,	28, 35,	28, 26, 5,	32, 28, 3,	26, 28,	26, 28,	32, 13, 6		28, 13,	32, 2	6, 28, 32	6, 28, 32	32, 35, 13	28, 6, 32	28, 6, 32	10, 26, 24	6, 19, 28, 24	6, 1, 32	3, 6, 32	
29	Manufacturing precision	28, 32,	28, 35,	10, 28,	2, 32, 10	28, 33,	2, 29, 18, 36	32, 23, 2	25, 10, 35	10, 28, 32	28, 19,	3, 35	32, 30, 40	30, 18	3, 27	3, 27, 40		19, 26	3, 32	32, 2	
30	Object-affected harmful factors	22, 21,	2, 22, 13, 24	17, 1, 39, 4	1, 18	22, 1, 33, 28	27, 2, 39, 35	22, 23,	34, 39,	21, 22,	13, 35,	22, 2, 37	22, 1, 3, 35	35, 24,	18, 35,	22, 15,	17, 1, 40, 33	22, 33,	1, 19, 32, 13	1, 24, 6, 27	10, 2, 22, 37
31	Object-generated harmful factors	19, 22,	35, 22, 1,	17, 15,		17, 2, 18, 39	22, 1, 40	17, 2, 40	30, 18,	35, 28, 3,	35, 28, 1,	2, 33, 27, 18	35, 1	35, 40,	15, 35,	15, 22,	21, 39,	22, 35, 2,	19, 24,	2, 35, 6	19, 22, 18
32	Ease of manufacture	28, 29,	1, 27, 36, 13	1, 29, 13, 17	15, 17, 27	13, 1, 26, 12	16, 40	13, 29, 1,	35	35, 13, 8,	35, 12	35, 19, 1,	1, 28, 13, 27	11, 13, 1	1, 3, 10, 32	27, 1, 4	35, 16	27, 26, 18	28, 24,	28, 26,	1, 4
33	Ease of operation	25, 2, 13, 15	6, 13, 1, 25	1, 17, 13, 12		1, 17, 13, 16	18, 16,	1, 16, 35, 15	4, 18, 39, 31	18, 13, 34	28, 13 35	2, 32, 12	15, 34,	32, 35, 30	32, 40, 3,	29, 3, 8, 25	1, 16, 25	26, 27, 13	13, 17, 1,	1, 13, 24	
34	Ease of repair	2, 27 35, 11	2, 27, 35, 11	1, 28, 10, 25	3, 18, 31	15, 13, 32	16, 25	25, 2, 35, 11	1	34, 9	1, 11, 10	13	1, 13, 2, 4	2, 35	11, 1, 2, 9	11, 29,	1	4, 10	15, 1, 13	15, 1, 28, 16	
35	Adaptability or versatility	1, 6, 15, 8	19, 15,	35, 1, 29, 2	1, 35, 16	35, 30,	15, 16	15, 35, 29		35, 10, 14	15, 17, 20	35, 16	15, 37, 1,	35, 30, 14	35, 3, 32, 6	13, 1, 35	2, 16	27, 2, 3, 35	6, 22, 26, 1	19, 35,	
36	Device complexity	26, 30,	2, 26, 35, 39	1, 19, 26, 24	26	14, 1, 13, 16	6, 36	34, 26, 6	1, 16	34, 10, 28	26, 16	19, 1, 35	29, 13,	2, 22, 17, 19	2, 13, 28	10, 4, 28, 15		2, 17, 13	24, 17, 13	27, 2, 29, 28	
37	Difficulty of detecting and measuring	27, 26,	6, 13, 28, 1	16, 17,	26	2, 13, 18, 17	2, 39, 30, 16	29, 1, 4, 16	2, 18, 26, 31	3, 4, 16, 35	30, 28,	35, 36,	27, 13, 1,	11, 22,	27, 3, 15, 28	19, 29,	25, 34, 6,	3, 27, 35, 16	2, 24, 26	35, 38	19, 35, 16
38	Extent of automation	28, 26,	28, 26,	14, 13,	23	17, 14, 13		35, 13, 16		28, 10	2, 35	13, 35	15, 32, 1,	18, 1	25, 13	6, 9		26, 2, 19	8, 32, 19	2, 32, 13	
39	Productivity	35, 26,	28, 27,	18, 4, 28, 38	30, 7, 14, 26	10, 26,	10, 35,	2, 6, 34, 10	35, 37,		28, 15,	10, 37, 14	14, 10,	35, 3, 22, 39	29, 28,	35, 10, 2,	20, 10,	35, 21,	26, 17,	35, 10,	1

	Worsening Feature → / Improving Feature ↓	Power	Loss of Energy	Loss of Substance	Loss of Information	Loss of Time	Quantity of substance	Reliability	Measurement accuracy	Manufacturing precision	Object-affected harmful factors	Object-generated harmful factors	Ease of manufacture	Ease of operation	Ease of repair	Adaptability or versatility	Device complexity	Difficulty of detecting and measuring	Extent of automation	Productivity
		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	Weight of moving object	12, 36,	6, 2, 34, 19	5, 35, 3, 31	10, 24, 35	10, 35,	3, 26, 18, 31	1, 3, 11, 27	28, 27,	28, 35,	22, 21,	22, 35,	27, 28, 1,	35, 3, 2, 24	2, 27, 28, 11	29, 5, 15, 8	26, 30,	28, 29,	26, 35 18, 19	35, 3, 24, 37
2	Weight of stationary object	15, 19,	18, 19,	5, 8, 13, 30	10, 15, 35	10, 20,	19, 6, 18, 26	10, 28, 8,	18, 26, 28	10, 1, 35, 17	2, 19, 22, 37	35, 22, 1,	28, 1, 9	6, 13, 1, 32	2, 27, 28, 11	19, 15, 29	1, 10, 26, 39	25, 28,	2, 26, 35	1, 28, 15, 35
3	Length of moving object	1, 35	7, 2, 35, 39	4, 29, 23, 10	1, 24	15, 2, 29	29, 35	10, 14,	28, 32, 4	10, 28,	1, 15, 17, 24	17, 15	1, 29, 17	15, 29,	1, 28, 10	14, 15, 1,	1, 19, 26, 24	35, 1, 26, 24	17, 24,	14, 4, 28, 29
4	Length of stationary object	12, 8	6, 28	10, 28,	24, 26,	30, 29, 14		15, 29, 28	32, 28, 3	2, 32, 10	1, 18		15, 17, 27	2, 25	3	1, 35	1, 26	26		30, 14, 7,
5	Area of moving object	19, 10,	15, 17,	10, 35, 2,	30, 26	26, 4	29, 30, 6,	29, 9	26, 28,	2, 32	22, 33,	17, 2, 18, 39	13, 1, 26, 24	15, 17,	15, 13,	15, 30	14, 1, 13	2, 36, 26, 18	14, 30,	10, 26,
6	Area of stationary object	17, 32	17, 7, 30	10, 14,	30, 16	10, 35, 4,	2, 18, 40, 4	32, 35,	26, 28,	2, 29, 18, 36	27, 2, 39, 35	22, 1, 40	40, 16	16, 4	16	15, 16	1, 18, 36	2, 35, 30, 18	23	10, 15,
7	Volume of moving object	35, 6, 13, 18	7, 15, 13, 16	36, 39,	2, 22	2, 6, 34, 10	29, 30, 7	14, 1, 40, 11	25, 26, 28	25, 28, 2,	22, 21,	17, 2, 40, 1	29, 1, 40	15, 13,	10	15, 29	26, 1	29, 26, 4	35, 34,	10, 6, 2, 34
8	Volume of stationary object	30, 6		10, 39,		35, 16, 32	35, 3	2, 35, 16		35, 10, 25	34, 39,	30, 18,	35		1		1, 31	2, 17, 26		35, 37,
9	Speed	19, 35,	14, 20,	10, 13,	13, 26		10, 19,	11, 35,	28, 32, 1,	10, 28,	1, 28, 35, 23	2, 24, 35, 21	35, 13, 8,	32, 28,	34, 2, 28, 27	15, 10, 26	10, 28, 4,	3, 34, 27, 16	10, 18	
10	Force (Intensity)	19, 35,	14, 15	8, 35, 40, 5		10, 37, 36	14, 29,	3, 35, 13, 21	35, 10,	28, 29,	1, 35, 40, 18	13, 3, 36, 24	15, 37,	1, 28, 3, 25	15, 1, 11	15, 17,	26, 35,	36, 37,	2, 35	3, 28, 35, 37
11	Stress or pressure	10, 35, 14	2, 36, 25	10, 36, 3,		37, 36, 4	10, 14, 36	10, 13,	6, 28, 25	3, 35	22, 2, 37	2, 33, 27, 18	1, 35, 16	11	2	35	19, 1, 35	2, 36, 37	35, 24	10, 14,
12	Shape	4, 6, 2	14	35, 29, 3,		14, 10,	36, 22	10, 40, 16	28, 32, 1	32, 30, 40	22, 1, 2, 35	35, 1	1, 32, 17, 28	32, 15, 26	2, 13, 1	1, 15, 29	16, 29, 1,	15, 13, 39	15, 1, 32	17, 26,
13	Stability of the object's composition	32, 35,	14, 2, 39, 6	2, 14, 30, 40		35, 27	15, 32, 35		13	18	35, 24,	35, 40,	35, 19	32, 35, 30	2, 35, 10, 16	35, 30,	2, 35, 22, 26	35, 22,	1, 8, 35	23, 35,
14	Strength	10, 26,	35	35, 28,		29, 3, 28, 10	29, 10, 27	11, 3	3, 27, 16	3, 27	18, 35,	15, 35,	11, 3, 10, 32	32, 40,	27, 11, 3	15, 3, 32	2, 13, 25, 28	27, 3, 15, 40	15	29, 35,
15	Duration of action of moving object	19, 10,		28, 27, 3,	10	20, 10,	3, 35, 10, 40	11, 2, 13	3	3, 27, 16, 40	22, 15,	21, 39,	27, 1, 4	12, 27	29, 10, 27	1, 35, 13	10, 4, 29, 15	19, 29,	6, 10	35, 17,
16	Duration of action by stationary object	16		27, 16,	10	28, 20,	3, 35, 31	34, 27, 6,	10, 26, 24		17, 1, 40, 33	22	35, 10	1	1	2		25, 34, 6,	1	20, 10,
17	Temperature	2, 14, 17, 25	21, 17,	21, 36,		35, 28,	3, 17, 30, 39	19, 35, 3,	32, 19, 24	24	22, 33,	22, 35, 2,	26, 27	26, 27	4, 10, 16	2, 18, 27	2, 17, 16	3, 27, 35, 31	26, 2, 19, 16	15, 28, 35
18	Illumination intensity	32	13, 16, 1,	13, 1	1, 6	19, 1, 26, 17	1, 19		11, 15, 32	3, 32	15, 19	35, 19,	19, 35,	28, 26, 19	15, 17,	15, 1, 19	6, 32, 13	32, 15	2, 26, 10	2, 25, 16
19	Use of energy by moving object	6, 19, 37, 18	12, 22,	35, 24,		35, 38,	34, 23,	19, 21,	3, 1, 32		1, 35, 6, 27	2, 35, 6	28, 26, 30	19, 35	1, 15, 17, 28	15, 17,	2, 29, 27, 28	35, 38	32, 2	12, 28, 35
20	Use of energy by stationary object			28, 27,			3, 35, 31	10, 36, 23			10, 2, 22, 37	19, 22, 18	1, 4					19, 35,		1, 6
21	Power	+	10, 35, 38	28, 27,	10, 19	35, 20,	4, 34, 19	19, 24,	32, 15, 2	32, 2	19, 22,	2, 35, 18	26, 10, 34	26, 35, 10	35, 2, 10, 34	19, 17, 34	20, 19,	19, 35, 16	28, 2, 17	28, 35, 34
22	Loss of Energy	3, 38	+	35, 27, 2,	19, 10	10, 18,	7, 18, 25	11, 10, 35	32		21, 22,	21, 35, 2,		35, 32, 1	2, 19		7, 23	35, 3, 15, 23	2	28, 10,
23	Loss of substance	28, 27,	35, 27, 2,	+		15, 18,	6, 3, 10, 24	10, 29,	16, 34,	35, 10,	33, 22,	10, 1, 34, 29	15, 34, 33	32, 28, 2,	2, 35, 34, 27	15, 10, 2	35, 10,	35, 18,	35, 10, 18	28, 35,
24	Loss of Information	10, 19	19, 10		+	24, 26,	24, 28, 35	10, 28, 23			22, 10, 1	10, 21, 22	32	27, 22				35, 33	35	13, 23, 15
25	Loss of Time	35, 20,	10, 5, 18, 32	35, 18,	24, 26,	+	35, 38,	10, 30, 4	24, 34,	24, 26,	35, 18, 34	35, 22,	35, 28,	4, 28, 10, 34	32, 1, 10	35, 28	6, 29	18, 28,	24, 28,	
26	Quantity of substance/the matter	35	7, 18, 25	6, 3, 10, 24	24, 28, 35	35, 38,	+	18, 3, 28, 40	13, 2, 28	33, 30	35, 33,	3, 35, 40, 39	29, 1, 35, 27	35, 29,	2, 32, 10, 25	15, 3, 29	3, 13, 27, 10	3, 27, 29, 18	8, 35	13, 29, 3,
27	Reliability	21, 11,	10, 11, 35	10, 35,	10, 28	10, 30, 4	21, 28,	+	32, 3, 11, 23	11, 32, 1	27, 35, 2,	35, 2, 40, 26		27, 17, 40	1, 11	13, 35, 8,	13, 35, 1	27, 40, 28	11, 13, 27	1, 35, 29, 38
28	Measurement accuracy	3, 6, 32	26, 32, 27	10, 16,		24, 34,	2, 6, 32	5, 11, 1, 23	+		28, 24,	3, 33, 39, 10	6, 35, 25, 18	1, 13, 17, 34	1, 32, 13, 11	13, 35, 2	27, 35,	26, 24,	28, 2, 10, 34	10, 34,
29	Manufacturing precision	32, 2	13, 32, 2	35, 31,		32, 26,	32, 30	11, 32, 1		+	26, 28,	4, 17, 34, 26		1, 32, 35, 23	25, 10		26, 2, 18		26, 28,	10, 18,
30	Object-affected harmful factors	19, 22,	21, 22,	33, 22,	22, 10, 2	35, 18, 34	35, 33,	27, 24, 2,	28, 33,	26, 28,	+		24, 35, 2	2, 25, 28, 39	35, 10, 2	35, 11,	22, 19,	22, 19,	33, 3, 34	22, 35,
31	Object-generated harmful factors	2, 35, 18	21, 35, 2,	10, 1, 34	10, 21, 29	1, 22	3, 24, 39, 1	24, 2, 40, 39	3, 33, 26	4, 17, 34, 26		+					19, 1, 31	2, 21, 27, 1	2	22, 35,
32	Ease of manufacture	27, 1, 12, 24	19, 35	15, 34, 33	32, 24,	35, 28,	35, 23, 1,		1, 35, 12, 18		24, 2		+	2, 5, 13, 16	35, 1, 11, 9	2, 13, 15	27, 26, 1	6, 28, 11, 1	8, 28, 1	35, 1, 10, 28
33	Ease of operation	35, 34, 2,	2, 19, 13	28, 32, 2,	4, 10, 27, 22	4, 28, 10, 34	12, 35	17, 27, 8,	25, 13, 2,	1, 32, 35, 23	2, 25, 28, 39		2, 5, 12	+	12, 26, 1,	15, 34, 1,	32, 26,		1, 34, 12, 3	15, 1, 28
34	Ease of repair	15, 10,	15, 1, 32, 19	2, 35, 34, 27		32, 1, 10, 25	2, 28, 10, 25	11, 10, 1,	10, 2, 13	25, 10	35, 10, 2,		1, 35, 11, 10	1, 12, 26, 15	+	7, 1, 4, 16	35, 1, 13, 11		34, 35, 7,	1, 32, 10
35	Adaptability or versatility	19, 1, 29	18, 15, 1	15, 10, 2,		35, 28	3, 35, 15	35, 13, 8,	35, 5, 1, 10		35, 11,		1, 13, 31	15, 34, 1,	1, 16, 7, 4	+	15, 29,	1	27, 34, 35	35, 28, 6,
36	Device complexity	20, 19,	10, 35,	35, 10,		6, 29	13, 3, 27, 10	13, 35, 1	2, 26, 10, 34	26, 24, 32	22, 19,	19, 1	27, 26, 1,	27, 9, 26, 24	1, 13	29, 15,	+	15, 10,	15, 1, 24	12, 17, 28
37	Difficulty of detecting and measuring	18, 1, 16, 10	35, 3, 15, 19	1, 18, 10, 24	35, 33,	18, 28,	3, 27, 29, 18	27, 40,	26, 24,		22, 19,	2, 21	5, 28, 11, 29	2, 5	12, 26	1, 15	15, 10,	+	34, 21	35, 18
38	Extent of automation	28, 2, 27	23, 28	35, 10,	35, 33	24, 28,	35, 13	11, 27, 32	28, 26,	28, 26,	2, 33	2	1, 26, 13	1, 12, 34, 3	1, 35, 13	27, 4, 1, 35	15, 24, 10	34, 27, 25	+	5, 12, 35, 26
39	Productivity	35, 20, 10	28, 10,	28, 10,	13, 15, 23		35, 38	1, 35, 10, 38	1, 10, 34, 28	18, 10,	22, 35,	35, 22,	35, 28, 2,	1, 28, 7, 10	1, 32, 10, 25	1, 35, 28, 37	12, 17,	35, 18,	5, 12, 35, 26	+