Application of TRIZ Principles In Software Concepts

Umakant Mishra
1. Introduction

The Inventive Principles of TRIZ are not only applicable to mechanical or technical field; they are also applicable to Software solutions. I have prepared a complete book on the application of each of the forty principles in software domain. This is one of my old articles while I was preparing for the book.

Most of the 40 principles are easy to learn, but the question arises where to apply and how to apply. This is very difficult to answer. But for the beginners first think of a problem and they try out solutions by applying the principles. Initially try out with known solutions. Some day the time may come that you explore an unknown solution.
2. Program coding:

- Divide the large project into smaller modules and sub projects (Principle-1: Segmentation).
- Do a planning and requirement analysis (Principle-10: Prior Action) before implementing the project.
- It’s advisable to prepare visual drawings and prototypes (Principle-27: Cheap and Disposable) before actual coding.
- Follow the coding standards and protocols for compatibility (Principle-6: Universal) with rest of the world.
- Break large routines (piece of codes) into small routines, functions, methods (Principle-1: Segmentation).
- Make the structure modular (Principle-17: Another dimension) to easily modify and restructure.
- Use comments (Principle-38: Enrich) adequately in between codes.
- Follow proper indenting and allow space (Principle-31: Hole) in between block of codes.
- Take out the repetitive codes (Principle-2: Taking out) and keep them in library routines.
- Small methods, functions are convenient for reuse (Principle-30: Thin and Flexible).
- Comment a suspicious piece of code while debugging and uncomment after trouble shooting (Principle-35: Parameter change, Principle-34: Discard and recover).
- It’s advisable to keep a copy of your source code (Principle-26: Copy) in safe place.
- A platform like VSS can be used to manage and protect source code (Principle-9: Prior counteraction ).
- Every user of VSS downloads a local copy (Principle-3: Local Quality) of the code to edit.
Follow naming conventions (Principle-6: Universality) to name the variables, methods, routines etc..


Avoid unused variables, fields or methods (Principle-2: Take out).

Use exception handlers (Principle-9: Prior Counteraction) and display customized error messages (Principle-3: Local quality, Principle-11: Cushioning).

3. **Website designing:**

Make your web pages light (Principle-30: thin & flexible) to get loaded quickly.

Change the contents periodically (Principle-19: Periodic action) to attract the visitors again and again.

Make the contents of the page resizable within the frames or tables (Principle-15: Dynamize) to adjust to different screen resolutions.

Keep the main topics on the home page and use hyperlinks (Nested Doll, Principle-7) to navigate from one page to other.

Display a “loading…” message (Principle-11: Cushioning) in while downloading movies and large files.

In case of a missing link, display some informative (Cushion, P-11) instead of a dissuasive error message.

Use colored fonts and backgrounds (Principle-32: Color Change) to make the web pages attractive.

Enrich the web page (Principle-38: Enrich) with graphics, animations, good contents and links.

Make your pages compatible to IE, Netscape and other browsers (Principle-6: Universality, Principle-33: Homogeneity/ Compatibility).

Include a feedback form (Principle-23: Feedback) to get visitors’ suggestions.
4. Java:

Java introduced a platform independent technology (Principle-6: Universality) into the field of software development.

The java code is first compiled to byte code (Principle-10: prior action) to finally run on a java virtual machine (JVM). The JVM interprets the byte code to the native machine code (Principle-36: State change/conversion).

JVM works in between java code and the physical machine (Principle-24: Intermediary).

The java virtual machine uses a method called garbage collection (Principle-34: Discard and Recover) to collect the unused memory periodically (Periodic Action, P-19).

5. OOP:

An object is a software bundle of related variables and methods (Principle-40: Composite).

Software objects are often used to model real-world objects you find in everyday life (Principle-26: Copying).

Software objects interact and communicate with each other using messages (Principle-24: Intermediary).

A class is a prototype for objects (Principle-6: Universality) that defines the variables and the methods common to all objects of a certain kind. The objects are created from classes (Principle-3: Local Quality).

A class inherits state and behavior from its super-class (Principle-26: Copying).

An interface is a collection of method and constant declarations (Principle-40: Composite).

An interface is used to communicate between the objects/classes (Principle-24: Intermediary).

When a class implements an interface, it promises to implement all of the methods declared in that interface (Principle-33: Homogeneity).
6. Networking:

The purpose of networking is to share (Principle-6: Universality) the resources in the network.

The machines are connected to each other (Principle-5: Merging) in a network.

The network cards, bridges, routers work as interface (Principle-24: Intermediary) to connect one device with another in the network.

The large networks are broken into small segments (Principle-1: Segmentation) and subnets.

Peer to peer network don’t use distinctive servers as every machine bears same responsibility (Principle-12: Equipotentiality).

Internet is the largest network consisting of many smaller networks and individual hosts (Principle-5: Merging).

Many hosts connect to internet and many hosts withdraw connection too at every moment from the internet (Principle-15: Dynamize).

Each machine or host has a name or IP address for easy identification in the network (Principle-3: Local Quality).

The data is transferred in small packets (Principle-1: segmentation) through the cable or wireless transmission media.

The sending device sends the packets continuously (Principle-20: Continuous Action) and the receiving device sends acknowledgements (Principle-23: Feedback) for verification.

7. Server:

A server is a machine which runs specific programs continuously as a service (Principle-20: Continuity of useful action).

A server can run many services simultaneously (Principle-5: Merging) like file service, web service, ftp service, email service etc. (Note: This will not be “Composite” as the services are all independent. But they run together on the same machine, hence “merging”)

Application of TRIZ Principles in Software Concepts by Umakant Mishra
A server Operating system is typically fault tolerant and capable of fixing of damages by itself (Principle-25: self service) in the normal course of function.

The clients interact with the server through a client connection program (Principle-24: intermediary).

The client should use the same protocol (Principle-33: Homogeneity/compatibility) to communicate with the server.

A server typically has separate volume (or partition or disks space) for system operation and separate volume for storing user data (Principle-1: Segmentation).

The user volume of a file server typically has separate home directories for each users (Principle-1: Segmentation).

The server gives users access to different resources based on their login authentication (Principle-3: Local Quality).

There are special purpose servers like Database server, Domain name server, DHCP server (Principle-3: Local quality) for different purposes.

8. Distributed Database:

Distributed database stores data in multiple different sites (Principle-1: Segmentation).

Local autonomy (Principle-3: Local Quality) allows each site to maintain an independent nature so data and resources can be protected and managed by local authorities.

Non-Centralization eliminates central data sites (Principle-1: Segmentation) that represent a single point of failure.

Partitioning provides a way to split the database and store it at multiple sites (Principle-1: Segmentation).

Transparency hides the location of the data (Principle-2: Taking out) from users so they don’t need to be concerned about the location of data.

Replication (Principle-26: Copying) provides a way to copy multiple fragments of the database to multiple sites.
Continuous operation (Principle-20: Continuity) provide services to users, even when the remote databases are not accessible or when the database is busy during backup.

Data updated at different places are synchronized periodically (Principle-19: Periodic Action).

9. Security:

Use password protection for data/system access (Principle-9: Prior Counteraction).

An initial login, although takes few seconds extra to get into the system, protects the system from unauthorized access (Principle-9: Prior Counteraction).

Periodically change your password (Principle-19: Periodic Action) to stay safe from hackers.

Use complex passwords comprising alphabets, numbers and special characters (Principle-4: Asymmetry)


Don’t connect to Internet or external network unless required (Principle-9: Prior counteraction). Disconnect from Internet when the work is over (Principle-2: Takeout).

If connected, use a firewall to separate your machine from outside world (Segmentation, P-1, Takeout, P-2).

Block all unused ports (Principle-2: Take out, Principle-31: Hole) to prevent unwanted inbound connections.

Use automode screen saver (Principle-25: Self Service) which activates within few seconds you leave the machine.


SSL uses data encryption (Principle-28: Mechanics Substitution) for data confidentiality and protection.
10. Fault Management:

Fault Management is the ability to locate faults, determine the cause and make corrections through the following.


- Setting threshold conditions (Principle-35: Parameter change) which can warn you with alarms (Principle-8: Counterweight) that may cause failures.

- Ensure uninstalling unwanted softwares (Principle-2: Taking out) that would reduce loan on your system.

- Implement fault tolerance (Principle-11: Cushioning) measures.

- Some Network Operating Systems (NOS) keep multiple copies (Principle-26: Copying) of their FAT (file allocation table) or file information data.

- Some network operating systems (NOS) provides hot fix feature (Principle-25: Self service) that automatically detects bad blocks on a disk and moves data from those areas to safe locations.

- Disk mirroring and duplexing provide fault tolerance (Principle-11: Cushioning). RAID (Redundant Arrays of Inexpensive Disks) provides uninterrupted data (Principle-11: Cushioning) even when a hard disk fails in the array.

- Using alternative communication channels (Principle-17: Another dimension) to provide backup for communication during media failure (Principle-11: Cushioning).

11. Data Backup:

- Data backup (Principle-26: Copying) is necessary for data recovery during disk crashes and emergencies (Principle-9: Prior Counteraction).

- The important data is copied (P-26: Copying) to an external storage media like data cartridge (Principle-2: Take out)

- A backup is advised to be taken regularly on every day (Principle-19: Periodic action).
The normal practice is to take a full backup on the first day and incremental backups (Principle-21: Skipping) on the other days of the week. This cycle is repeated again for subsequent weeks and months (Principle-20: Continuous action).

The backup media is preserved for a period of one month to 6 months depending the nature of data. Then the old cartridges are recycled (Principle-34: Discard and recover) to take new backups.

Often a compressed backup (Thin, Principle-30) is preferred to save the backup space.

Most backup utilities verify (Principle-23: Feedback) the correctness of the backup as a part of the backup process.

As the system volume is not updated by users, it’s not backed up regularly (Principle-21: Skipping).

The backup administrator can schedule an automatic backup (Principle-25: Self Service) in which case he need not be physically present at the system.

The scheduled backup is normally scheduled to take place during nights when the server is relatively free (Principle-20: Continuous action).

12. Boosting PC Performance

Prevent PC crashes by avoiding untested and unreliable software (Principle-9: Prior Counteraction).

Optimize usage of computer resources (Principle-20: Continuity of useful action).

Auto recovery of Memory that Windows steals from your computer. (Principle-13: Reversing, Principle-25: Self Service)

Clean hundreds of unwanted garbage files from your hard disk. (Principle-2: Taking out)

Add/remove Windows Start Menu Options. Customize your default 'My Documents' folder, remove 'Favorites', 'Windows Update' options and more customizations. (Principle-28: Mechanics substitution, Principle-17: Another Dimension)
Enable Fast Shutdown and auto close of non-responding applications. (Principle-21: Skipping)

Refer to real time memory usage monitoring graph in task manager (Principle-23: Feedback).

CPU Stability Test to make sure that your system is stable. (Principle-23: Feedback)

Use a hang manager to rescue data during system crashes and hangs (Principle-11: Cushioning)

13. Conclusion

The deeper we analyse on any of these software concepts the more we find the applicability of inventive principles. Some people may ask, the above solutions are all known, what is the use of trying out the Principles on them. My answer is it is like learning a bicycle. If you are capable of riding the cycle on a known road, there will be a time when you can successfully ride the cycle on an unknown land. Our persuasion will continue in subsequent articles to review more of the TRIZ techniques for Software solutions.

Reference

