LEAN CARDS

POKA-YOKE

Mistake - Proofing

Eliminate product defects by preventing, correcting, or drawing attention to human errors as they occcur.

In Japanese, it means mistake proofing."

Ideally, products should be designed in a way it makes impossible to use them the wrong way.

Ensure that the right conditions exist before a process step is executed, and thus preventing defects from occurring in the first place. Where this is not possible, Poka Yokes perform a **detective function**, eliminating defects in the process **as early as possible**.

#error #mistake #proofing #prevent #defect #poka #yoke

HOW TO

- Poka-Yokes must be **implemented early** in the development process, so they can provide feedback quickly.
- They should be **precise** to easily diagnose and identify the mistake.
- They should be **simple to develop and maintain**: complex things have a high probability to fail. Having an erroneous Poka-Yoke is worse than not having one.
- Poka-Yokes have to be **transparent to people**: they should not be an obstacle to the developer or the users. Imagine having to run an hour-long test before committing your code changes!

EXAMPLES

For end-users

- Any software that implements an **auto-save** feature.
- Sending an email with no subject or body produces a **warning** that stops you from a possible mistake. It also warns you if there is no file attached and the email contains the word "attached".

For developers

STRONGLY RELATED TO JIDOKA !

- **Unit tests** and **pre-commit scripts**.
- **IDE's** indicate issues in your code and mistakes before even compiling.
- **Password strength indicators** prevent you from using weak passwords.
- **Double entry boxes** ask you to write the same critical value twice to prevent you from making a fatal mistake.