

# Concept: Project

## On this page:

- [Definition](#)
- [Relation to workflow and product model](#)
- [Re-using of projects](#)

## Related articles

- [Concept: User](#)
- [Concept: Property](#)
- [Concept: Task](#)
- [Concept: Widget](#)
- [Concept: Part](#)

## Definition

---

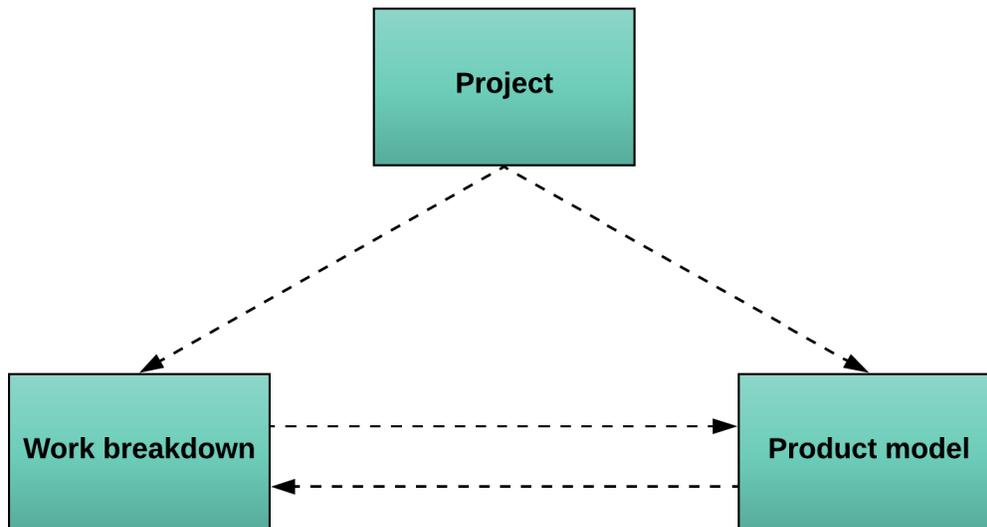
**Projects** are the main container for work in **KE-chain**, mirroring projects in the real world. In KE-chain, a project is executed by a project team consisting of application **users**. A project is made out of two fundamental components: the **work breakdown** and the **data model**. These two are closely related: by performing the **tasks** within the work breakdown the **product in the explorer** is created.

The result of a project is a fully defined product according to its data model definition and the user inputs. The finished product can then be used to provide information to the customer. This information can be supplied in multiple ways, such as the KE-chain task forms themselves, the KE-chain API or downloadable figures and reports.

## Relation to workflow and product model

---

Functionally speaking, one project is always coupled to one work breakdown and one data model. Within the project, the **parts** and **properties** considered for a product are created in the **data model**. Specific values of these parts and properties are defined during tasks listed under the **work breakdown**.



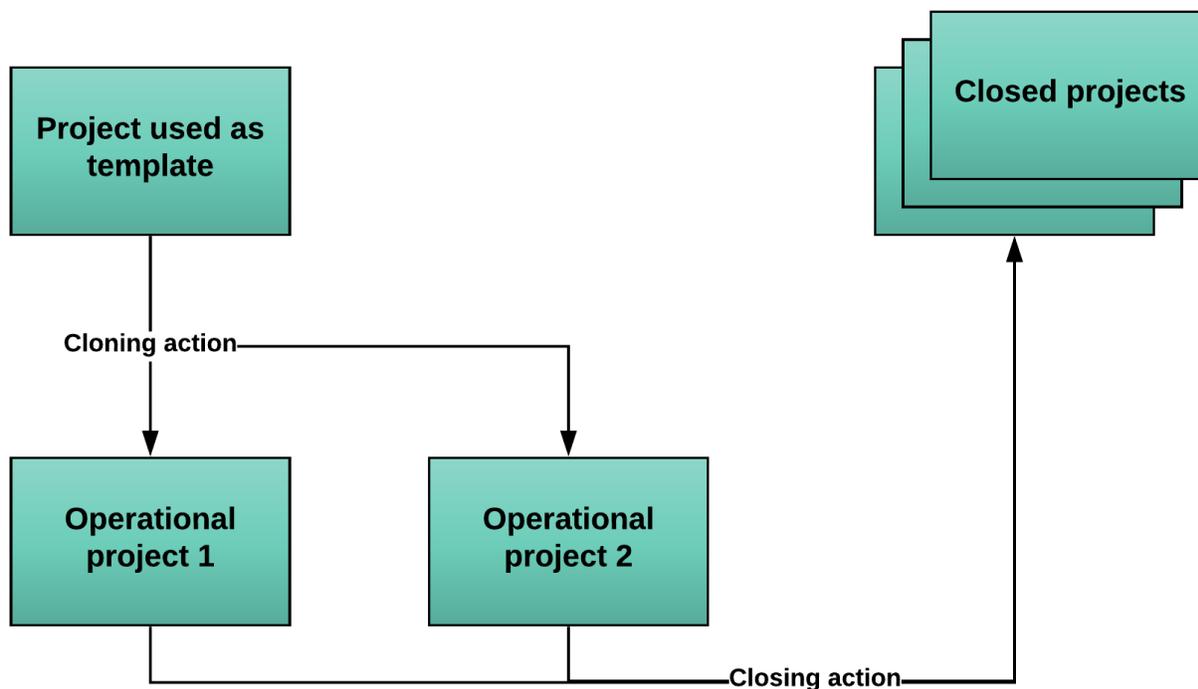
## Re-using of projects

---

KE-chain can also use **projects** as templates to jump-start any project. A project can have only its product model and work breakdown configured. Clones can be created based on these template projects and from that point, the product will be independently developed based

on the project's purpose. Therefore, 2 or more projects can have the same product model and (optionally) same work breakdown, but will deliver different results. Subsequently, the template project can be re-used an arbitrary number of times to create operational projects. Within an operational project, the end users (mostly engineers) will perform work. By executing and completing **tasks**, they arrive at a final **product**, completing a project. Once a project is completed, it is **closed**.

The figure below shows a diagram of this **project lifecycle**.



**How do I know which one of the projects is my template?**

Make sure you name your **project templates** accordingly, such that no confusion can be made between the **templates** and the **operational** projects. Also, it is useful to specify in the naming or description of the operational project the **template** it is based on, as it may be difficult to keep track of your projects otherwise.