

Der ingenieurmäßige Gestaltungsprozess

Version 1.0

Das Projekt ITAKE lehnt sich stark an den von Seymour Papert entwickelten **Konstruktionismus** an, einer Weiterentwicklung des Konstruktivismus, der auf Jean Piaget zurückgeht.

Papert ging es ursprünglich gar nicht unbedingt darum, dass Kinder Programmierer werden, sondern dass sie eine Umgebung angeboten bekommen, in der sie bedeutsame Ideen explorieren und dabei lernen können. Es geht in erster Linie darum, wie Kinder am besten lernen für eine ständig sich verändernde Welt. Papert greift dabei auch auf den Begriff Mathetik zurück: ... *möchte ich das Substantiv Mathetik für eine Unterweisung in der Kunst des Lernens verwenden.*'

In den USA spricht man in diesem Zusammenhang von einem **Engineering Design Process**, der dem Ganzen zu Grunde liegt und der schon von früh auf den Kindern vermittelt wird. Es ist nicht leicht, eine adäquate deutschsprachigen Übersetzung zu finden. Nach eingehenden Recherchen hat sich „**Der ingenieurmäßige Gestaltungsprozess**“ als am besten geeignet erwiesen.

Interessante Unterlagen findet man dazu z.B. beim Museum of Science in Boston:

http://legacy.mos.org/etf/etf_tg_introduction.pdf

Das Museum of Science hat dafür einen kindergerechten Prozess entwickelt:

http://teched.vcsu.edu/cmsfiles/46/mos_vcsu_wvqw21.pdf?1373544087

Dieser kindergerechte „Engineering Design Process“ besteht aus 5 Schritten:

http://www.teachersdomain.org/assets/wgbh/eng06/eng06_doc_lpaengprocess/eng06_doc_lpaengprocess.pdf

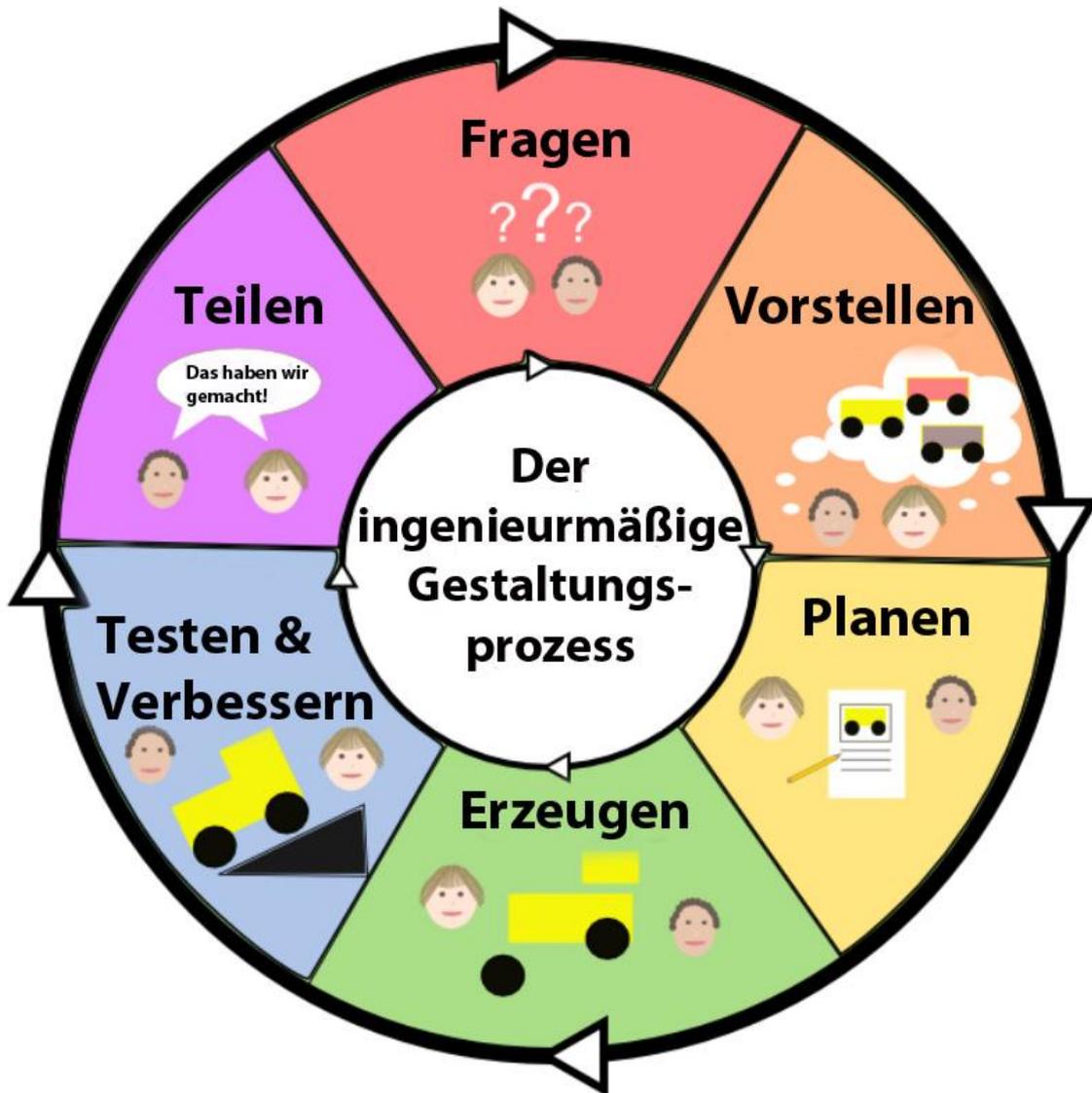
Die Tufts University hat den weiteren Schritt des „Teilens“ an das Ende dieses Prozesses angefügt. Dieses „Teilen“ gehört bei der Programmiersprache SCRATCH zu den wichtigen Komponenten, die den Lernerfolg zusätzlich fördern (Online-Plattform der entwickelten Projekte).



<http://creativecommons.org/licenses/by-nc-sa/4.0/>

www.coding4you.at

Im Folgenden haben wir versucht, die englischsprachige Version der Tufts University in eine deutsche Version überzuführen.



Quelle des Originals:

A Curriculum Unit on Programming and Robotics

Prof. Marina U. Bers, Louise Flannery, Elizabeth Kazakoff, and R. Jordan Crouser

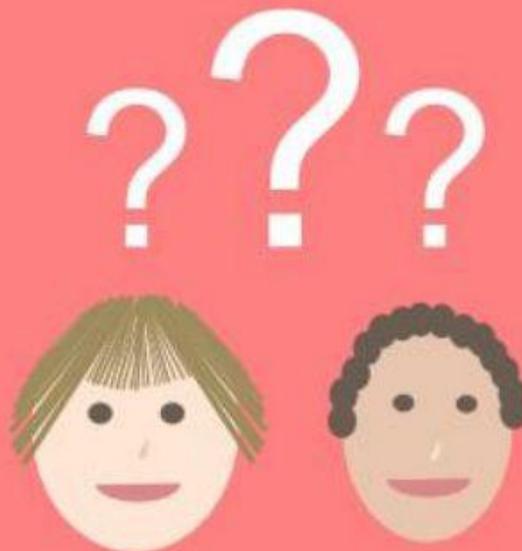
DevTech Research Group



<http://creativecommons.org/licenses/by-nc-sa/4.0/>

www.coding4you.at

Fragen



Man stellt sich Fragen zu einem Problem, das gelöst werden soll oder zu einem Ziel, das man erreichen möchte:

Was ist das Problem?

Wie haben andere dieses Problem gelöst?

Wie schauen die Rahmenbedingungen aus?

Vorstellen

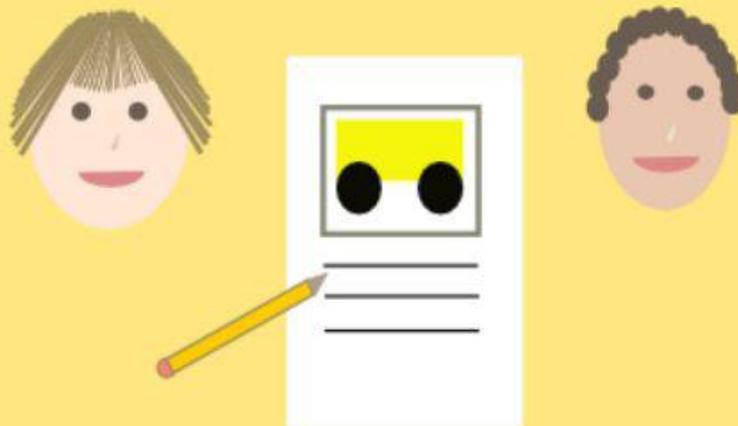


Stell die möglichst viele verschiedene Lösungsmöglichkeiten vor:

Welche Lösungen gibt es?

Für ein Brainstorming zur Ideengewinnung durch

Planen



Wähle eine der Lösungsmöglichkeiten aus und arbeite einen detaillierten Plan aus.

Zeichne ein Diagramm.

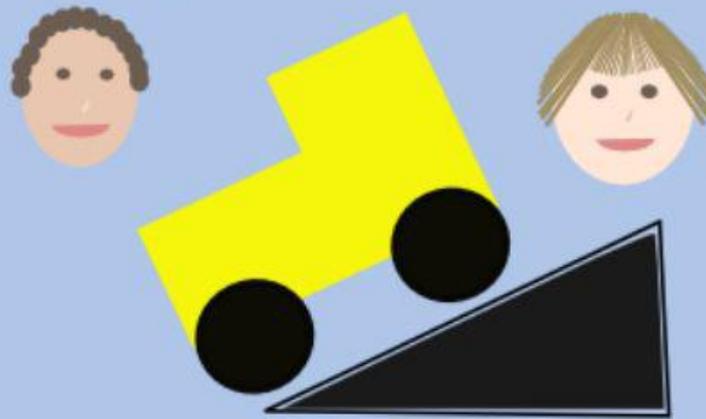
Stelle eine Liste von Materialien zusammen, die benötigt werden.

Erzeugen



Erzeuge einen Prototyp oder eine Arbeitsversion des Planes.

Testen & Verbessern



Test dein Produkt und beobachte, wie gut es mit deinen Zielen übereinstimmt.
Versuche verschiedene Verbesserungsmöglichkeiten und überprüfe sie.

Teilen

**Das haben
wir gemacht!**



Teile deine Ergebnisse anderen mit und erhalte von ihnen Rückmeldungen (Feedback).