

Mobile Computing From Prototype to Connected Product

CC BY-SA, T. Amberg, FHNW
(unless noted otherwise)

Slides: tmb.gr/mc-pro



[Source](#) on GDocs

Overview

These slides show *how prototypes become products*.

How to get from an idea to a physical prototype.

Scaling up from small batch to mass product*.

*Or at least some insight on what it takes.

Prerequisites

We'll use [OpenSCAD](#) software to create a 3D design.

And [Cura](#), a tool to prepare a 3D design for printing*.

*No hardware is required for this lesson.

Building connected products

So far we focused on firmware and backend software.

To create a real product, there are additional steps*:

Ideation → Prototyping → Development → Testing →

Production → QA → Logistics → Marketing → Sales →

Operations → Support → Maintenance → Sunsetting.

*Vendor perspective, there might be iterations.

Ideation

There are methodologies to find product ideas.

E.g. **Know Cards** by Alex Deschamps-Sonsino.

Or the **IoT Design Kit** by Studio Dott in Belgium.

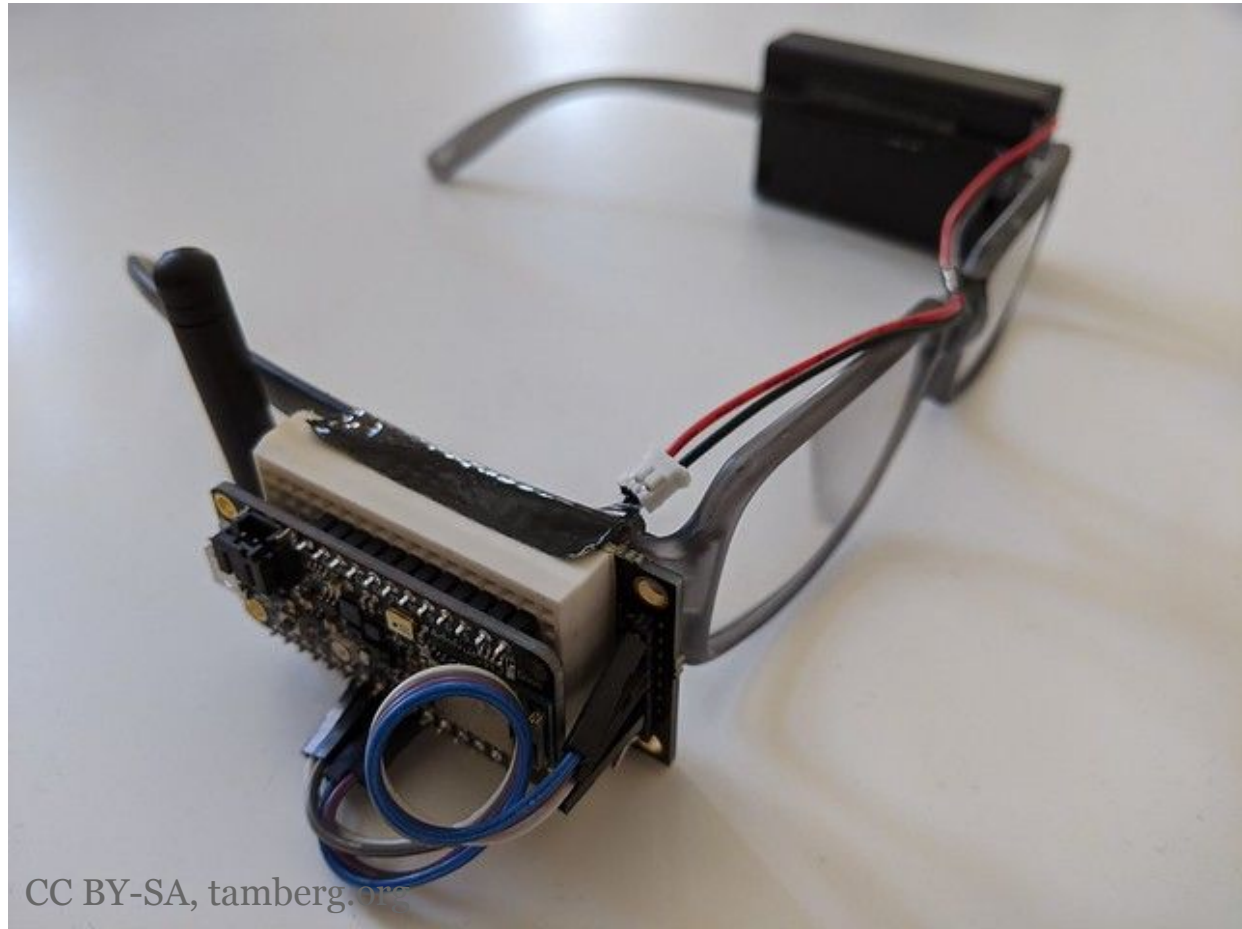
Or **Loaded Dice** by Albrecht Kurze, TU Chemnitz.

Attending hackdays is another way to get new ideas.

Hackdays

Assemble a few people w/ complementary skill sets and roles.

Let them build whatever they can imagine.



Prototyping

There are roughly these three variants of prototypes:

Design prototype, how the device will look and feel.

Functional prototype, device/service user experience.

Technical prototype, real hardware, often still too big.

These prototypes can be developed in parallel.

Prototypes

Left: Functional prototype, most electronics are in the base.

Right: Design prototype, no electronics.



Product development

Software / hardware are often developed in parallel.

Technical prototypes are miniaturised, integrated.

Functional prototypes are re-implemented.

Design prototypes are productised.

Testing and certification

Software and hardware, unit and integration testing.

Field studies, with representative (beta) user groups.

Certification, for each new HW version of a product.

QA at the factory for every instance of a device type.

Monitoring backend services to guarantee SLA.

Important IoT system qualities

Security, to keep devices, network & backend secure.

Privacy, to keep people in control of their own data.

Interoperability, to become part of an ecosystem.

Openness, standards & open source build trust.

See, e.g. [betteriot.org principles](https://betteriot.org/principles) for guidance.

Prototyping at a Fab Lab

A shared workshop for personal digital fabrication.

Computer-controlled tools to make almost anything:

3D printers, laser cutters, CNC mills, electronics lab.

Coming from MIT, [Fab Labs](#) are a global movement.

Made in a Fab Lab => can be replicated in others.

3D printing

Additive manufacturing, builds objects layer by layer.

Fused deposition modeling (FDM) is commonly used.

Filament materials include PLA and ABS (like Lego).

Stereolithography (SLA) is great for very small parts.

Selective laser sintering (SLS) is rather expensive.

Many schools have printers, e.g. [FHNW](#), [ZHAW](#).

3D printing process

Create a 3D model with CAD software, export as STL.

Printer-specific slicer* cuts the STL model into layers.

Resulting GCODE file is transferred to the 3D printer.

Printing takes 10-s of minutes up to multiple hours.

E.g. Cura, Slic3r or Prusa Slicer.

CAD software

CAD (computer aided design) tools for 3D modelling:

Commercial tools, e.g. [Onshape](#), [Rhino](#), [Solidworks](#).

Some are free to use for students, e.g. [Fusion 360](#).

Open source tools, e.g. [Blender](#), [OpenSCAD](#).

All of these can export STL files.

Hands-on, 15': Parametric design

Download [OpenSCAD](#) and create a simple 3D design.

OpenSCAD is a domain specific [language for CAD](#).

Objects can be built by [subtracting](#) simple shapes.

Design a box that fits a [Raspberry Pi Zero](#).

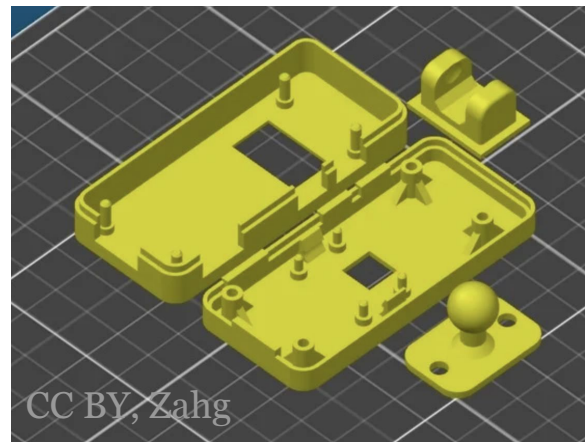
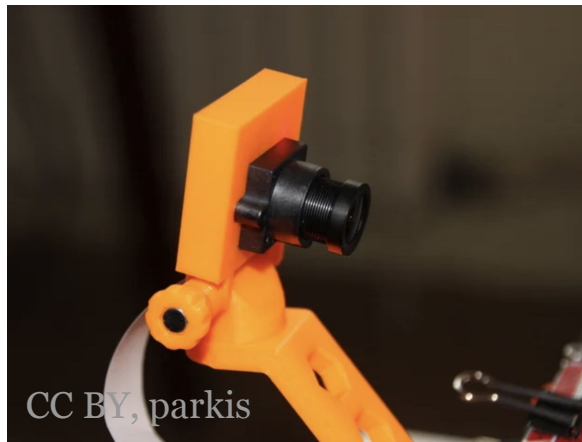
Export as STL and slice it e.g. in [Cura](#).

Done? Take a look at [this model](#).

Design repository

E.g. [Thingiverse](#), for openly licensed 2/3D models.

Here's a selection of Raspberry Pi camera adapters.



Laser cutting

Create a 2D design with vector based CAD software*.

Lasers know vector file formats like AI, DXF or PDF.

Materials include wood and acrylic sheets, < 10 mm.

Power/speed settings depend on material/thickness.

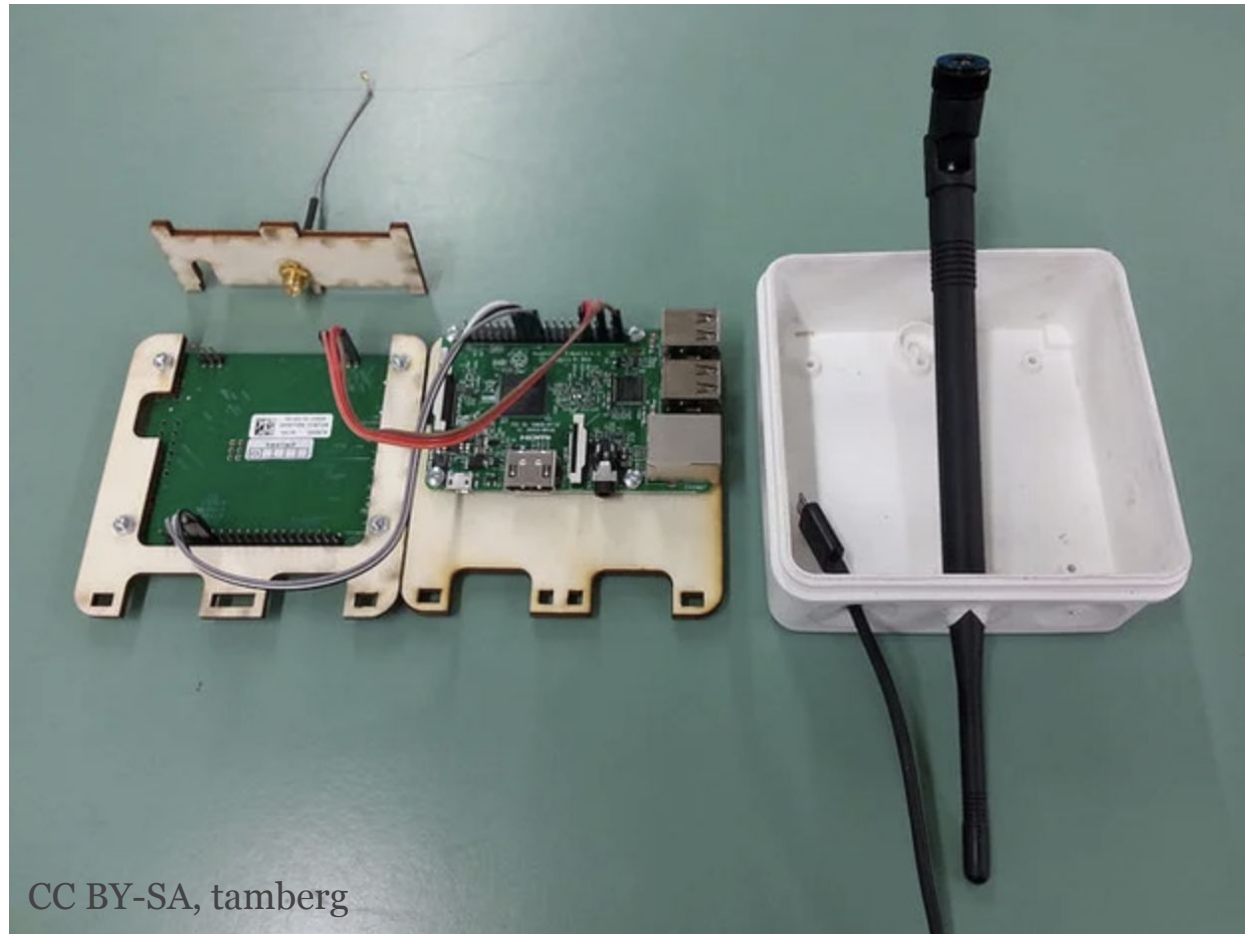
Laser cutting is fast, usually just takes a few minutes.

*E.g. [Inkscape](#), a free and open source design tool.

Laser-cut adapters

A quick way to fit electronics into an existing enclosure.

This is a simple **LoRa gateway**.



CNC milling

Subtractive manufacturing, cuts object out of a block.

3 or more axes, working area from 10-s of cm up to m.

2D or 3D CAD model is prepared with CAM software.

Materials include PCB, foam and wood, metal is hard.

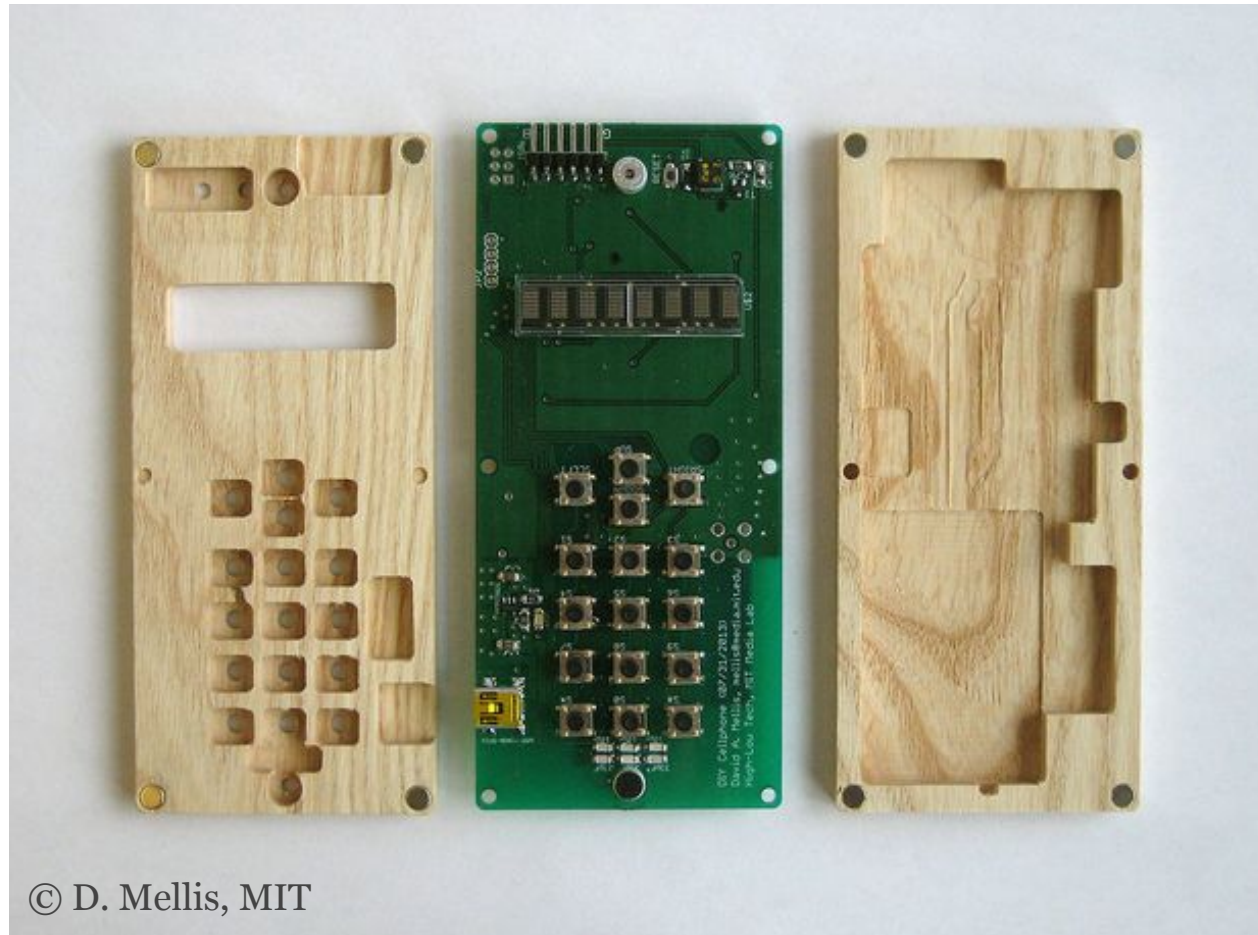
Tool head type and size must be taken into account.

Basic CNC milling can be learned in about a day.

CNC milled enclosure

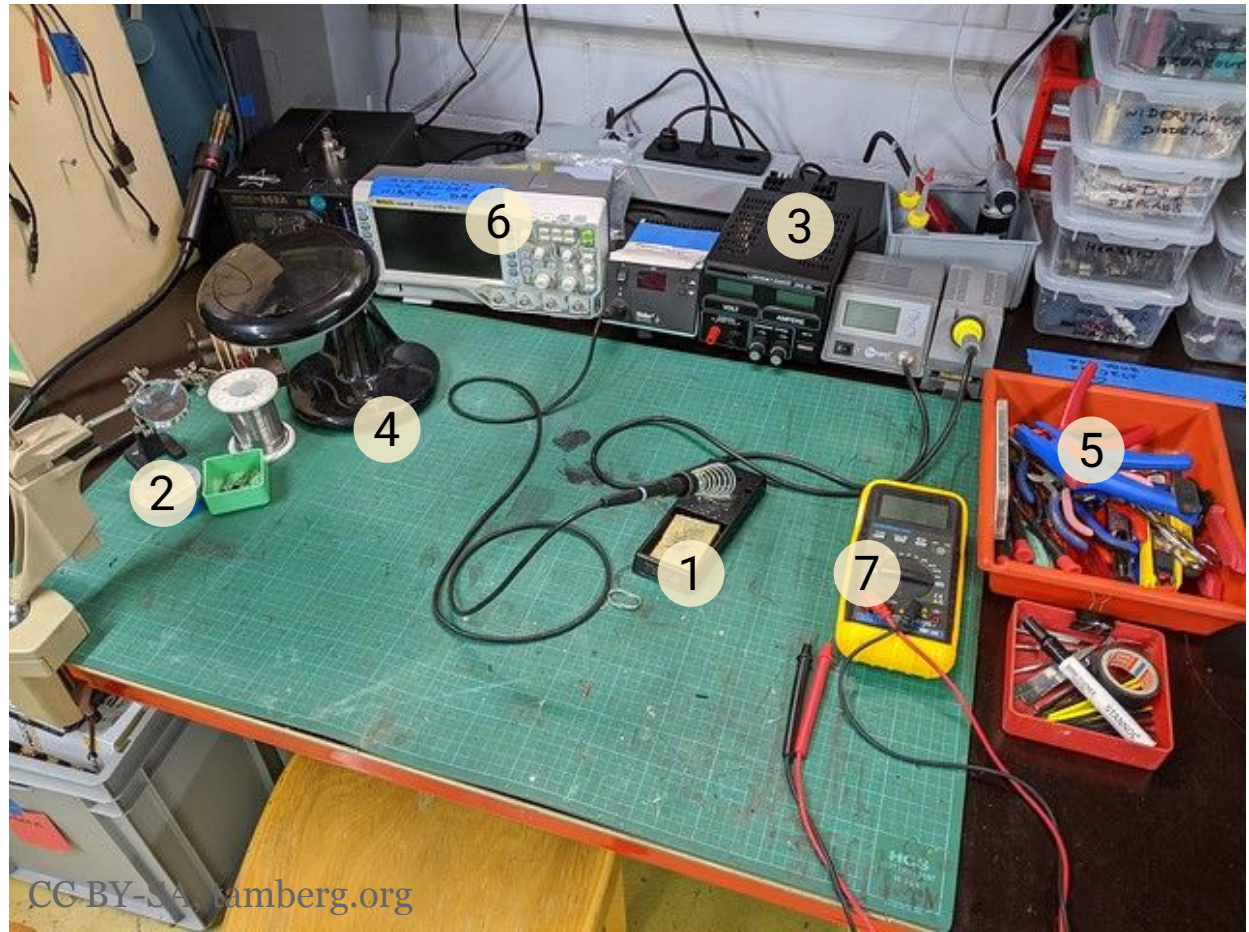
Nice materials
for prototypes,
too expensive in
production.

This is a simple
DIY cell phone.



Electronics laboratory

Soldering iron¹,
helping hands²,
power supply³,
lamp⁴, pliers⁵,
oscilloscope⁶,
multimeter⁷.



Designing a custom PCB

A printed circuit board (PCB) helps to integrate parts.

E.g. a controller, sensors, a radio & a battery holder.

The traces on a PCB are designed with a layout tool.

Electronic parts come with ready to use footprints.

Layout software includes [Fritzing](#), [Eagle](#) or [Kicad](#).

Producing a custom PCB

Single or double sided PCBs can be etched "by hand".

But it's easier and quite fast to get your PCBs made.

E.g. [Aisler](#), [OSHPark](#), [PCBWay](#) make small batches.

Electronic components can then be hand soldered.

Sourcing parts

The list of all parts is called *bill of materials* (BOM).

Shops for makers include [Adafruit](#), [Sparkfun](#) & [Seeed](#).

Electronic parts suppliers are, e.g. [Digikey](#) & [Mouser](#).

Or platforms like [AliExpress](#) and [Taobao](#) in China.

And for large batches, talk to manufacturers.

Prototype product

Focused on quick development, OK for beta users.

Off-the-shelf components, easy to use modules.

Off-the-shelf box, or 3D printed enclosure.

Custom PCB, manual assembly/soldering.

Batch size 1 to 100.

Small batch product

Focused on getting a certified product to real users.

Off-the-shelf, certified modules to save one-off cost.

Off-the-shelf or custom injection molded enclosure.

Custom PCB, automated assembly. Manual QA.

Batch size 100 to a few 1000.

Mass product

Optimised for cost, ease of use to reduce support.

Custom, certified PCB with integrated modules.

Custom injection molded parts, ~\$10k per mold.

Automated assembly and more automated QA.

Batch size over 10'000.

Automated assembly

A pick & place machine puts parts on a PCB which are then soldered in an oven.



Product examples

Opening products is a great way to learn about them:

[Avelon Wisely](#) LoRaWAN temp. and humidity sensor.

[Gardena smart gateway](#) from [Lemonbeat](#) to Internet.

A [TTN indoor gateway](#) and the [Belkin WeMo switch](#).

[Nest learning thermostat](#) and [Amazon Echo](#) device.

Wisely

MCU, radio
module and
sensors are
integrated
on the PCB.

Off-the-shelf
enclosure.



CC BY-SA, tamberg.org

Gardena gateway

Custom PCB
with existing
Linux SoC¹ and
radio module².

Custom plastic
enclosure.



CC BY-SA, tamberg.org

TTN indoor gateway

Encapsulated power supply¹.

Connector² for Wi-Fi module, probably not yet cost optimised.



Belkin smart plug

Separate PCB
for high voltage
and Wi-Fi part.

Cost optimised.

See [tear-down](#).



CC BY-NC-SA iFixit

Nest

Optimised for installation by customer.

Built-in level, spring-loaded wire terminals.

See [tear-down](#).



CC BY-NC-SA iFixit

Echo

Optimised for audio quality.

Many, complex
plastic parts.



See [tear-down](#).

Hands-on, 5': Second gen products

Compare the **Echo Dot (2nd gen)** to the original **Echo**.

What was changed, and what could be the rationale?

If an injection mold is \$10k, how much was saved?

Which other parts could bring down the price?

Lean startup methodology

Described by Eric Ries, in his book [The Lean Startup](#):

Build, measure, learn, to discover product/market fit.

Minimum viable product, to learn about customers.

Pivot, to change course, test a new hypothesis.

This process iterates towards a working business.

MVP

A minimum viable product, made from off the shelf parts.

VeloTracker.ch
connected bike
tracker / light.












CC BY-SA, tamberg.org

Business model canvas

A tool to prototype business models.

Get it [here](#).

The Business Model Canvas		Designed for:	Designed by:	Date:	Version:		
Key Partners  <small>Who are our Key Partners? Who are our key suppliers? Which Key Resources are we acquiring from partners? Which Key Activities do partners perform?</small> <small>CONSEQUENCES FOR INVESTMENT: Specialization and learning Reduction of risk and uncertainty Exploitation of particular resources and activities</small>	Key Activities  <small>What Key Activities do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue streams?</small> <small>CONSEQUENCES: Process learning Platform business Platform business</small>	Value Propositions  <small>What value do we deliver to the customer? Which one of our customer's problems are we helping to solve? What bundles of products and services are we offering to each Customer Segment? Which customer needs are we satisfying?</small> <small>CONSEQUENCES: Markets Relationships Channels Distribution "Turning the job over" Brand/Label Price Cost Reduction New Revenue Accessibility Convenience/Usability</small>	Customer Relationships  <small>What type of relationship does each of our Customer Segments expect us to establish and maintain with them? Which ones have we established? How are they integrated with the rest of our business model? How costly are they?</small> <small>CONSEQUENCES: Personal assistance Self-Service Automated Personal Assistance Self-Service Community Co-creation</small>	Customer Segments  <small>For whom are we creating value? Who are our most important customers?</small> <small>Market Model: Mass Market Segmented Niche Market Dual-sided Platform</small>			
	Key Resources  <small>What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue Streams?</small> <small>CONSEQUENCES: Platform Platform business Platform business, supporting, 3rd Party Platform</small>		Channels  <small>Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones work best? Which ones are most cost efficient? How are we integrating them with customer outcome?</small> <small>CONSEQUENCES: A. Accessible How do we use experience about our company's products and services? B. Scalable How do we help customers make our organization's Value Proposition? C. Affordable How do we allow customers to purchase goods, products and services? D. Fast How do we deliver a Value Proposition to customers? E. Other sales How do we provide cost customer support?</small>				
Cost Structure  <small>What are the most important costs inherent in our business model? Which Key Resources are most expensive? Which Key Activities are most expensive?</small> <small>AS YOUR BUSINESS GROWS: Cost drivers: Material cost, structure, how often value proposition, platform activities, network externalities Value Chain: Network on value creation, platform value proposition</small> <small>CONSEQUENCES: Platform business, 3rd Party Platform business Platform business Platform business Platform business</small>		Revenue Streams  <small>For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How would they prefer to pay? How much does each Revenue Stream contribute to overall revenues?</small> <table border="0"><tr><td>Types: Subscription Usage Fee Lump Sum Freemium Advertising Commission Brokerage Fee Licensing Rental Royalty</td><td>Direct Revenue: License Fee Product/Service Subscription Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee</td><td>Indirect Revenue: Advertising (Sponsorship) Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee</td></tr></table>			Types: Subscription Usage Fee Lump Sum Freemium Advertising Commission Brokerage Fee Licensing Rental Royalty	Direct Revenue: License Fee Product/Service Subscription Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee	Indirect Revenue: Advertising (Sponsorship) Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee
Types: Subscription Usage Fee Lump Sum Freemium Advertising Commission Brokerage Fee Licensing Rental Royalty	Direct Revenue: License Fee Product/Service Subscription Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee	Indirect Revenue: Advertising (Sponsorship) Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee Usage Fee					

Summary

We saw the steps involved from prototype to product.

We looked at digital fabrication as a prototyping tool.

We learned about PCBs, layout software, pick & place.

We got some insight into production at various scales.

This was the last lesson before the project kick-off.

Feedback or questions?

Join us on [MSE TSM MobCom](#) in MS Teams

Or email thomas.amberg@fhnw.ch

Thanks for your time.