Why digital products fail:
missplaced priorities
igorance about real users
conflicts of interrests
lack of design process

Goals: Expection of an end-condition, not only what a user does, but why.

Types of Goals:
End Goal (efficency): we want to get sth. done
Experience Goals (userexperience): we want to feel sth.
Life Goals (long term): what we ant to be

Non-user-goals:
Customers goals, Bussines goals, Technical Goals

Goal orientates design process:
Research: behaviro, patterns, user goals
Modeling: personas, narratives
Requirements: „What kind of fkt. we need?„
Framework: define Form, elements, key-path, grouping
Refinement: sketching, datail the layout
Development support: support & oversee delopens

tasks, activities, tasks and functionalities helping to reach a Goal:
Activities are composed of tasks
Tasks are composed of actions
Actions are composed of operations/ functionalities

A Goals is supportet by Activities
A Activity is made out of Tasks
A Tasks is out of Functionalities

Example: Coffee Machine Selling Site
Goal: The user want to have a cup of Coffee every Morning
Taks: The user should choose and buy a coffee Machine easily
functionalitites: ordering by pricing fkt., easy payment procedure, price comparison fkt.

Implementations Model
- representation how a machine or application works

Mental Model
- Mental models is how you think something will work based on your learnings & experience.

Interaction Framework:
- Form Factor: interface, physical device
- Posture: use short or long time use
- Input/ Output: what the user put in/ out the system
- Data Elements: ex. pdf-documents
- Functional elements: which fkt. are important or less important
- Grouping Fkt & layout: Positioning of the Elements
- Key Path senarios: sequens of interactions

Sketching: Sketching how many ideas as possible
Design Validation: user testing
refinement: look & feel

Why usability test
- It helps to recognize if you forgotten anything.
- Explore on usual cases
- Breaks down; What if, How does it look, How does it work for me.
- For user testing first; Paper Prototype and next Screen Based.
- go back and use what worked and do it again

Drawbacks
- dont test for long terms
- to test the affordance of a export user is difficult

Formative usability testing
- doing on the fly during the development, to find out if I am on the right track
- to do early enough to allow adjustments
- aspects of user experience approach

Summative usability testing
- with actual user
- moderator is a non-designer
- near the end of a dev. process

4 values, we should design for
- Ethical (considerate, helpful, Do no harm)
- Purposeful (useful, usable)
- Pragmatic(viable,feasible)
- Elegant (efficient, artful, affective)

Ethical interaction design
- Personal harm (loss of dignity, insult, humiliation)
- Psychological harm (confusion, discomfort, frustration, coercion, boredom)
- Physical Harm (pain, injury, deprivation, death, compromised safety)
- Economic Harm (loss of pro ts, loss of productivity, loss of wealth or savings)
- Social Harm (exploitation, creation, or perpetuation of injustice)
- Environmental Harm (pollution, elimination of biodiversity)

Chapter 08 - Digital etiquette

“taking an interest (Interesse)”
- Software should behave like a considerate human being.
- software should remember in our habits

“being conscientious (gewissenhaft)”
- The computer does the work, and the person does the thinking.
- Do not burden the user, do not ask again and again

do not complain (klagen), do not make unnecessary announcements

“failing gracefully”
Fail gracefully — no bizarre error messages

“bending the rules”
Allow for alternate ways to do things the bending of rules and undoing of mistakes, undo instead of verify

“taking responsibility”
The software should be smart enough to handle upcoming results from the actions of the user. Example printer send pages to the printer buffer, after pressing “cancle” the printer still printing a few pages.

Chapter 09 - Platform and posture

sovereign posture
- Large applications
- full-screen applications
- in use for long preiod of time (30min/1h/1day/...)
- work critical tasks (video editing, creating content)
- Examples: MS Word, Sketch up
- Rich toolbars: can have more than one
- Rich inputs: functions can be done in different ways
- Rich modeless feedback
- direct manipulation

optimized for intermediate user, not aimed on beginners
there should feature a conservative visual style

Transient posture
- acts in the moment
- temporary nature
- it’s used for small or singular tasks
- not often used, short use
- Example: Video Recording App, Mac Calculator

sometimes visual boring but effective, or more bold and more attention of the user (no possibilities of confusion or mistakes)
providing single functionalities with some accompanying controls
performs its job, avoke if it’s needed.
Transient applications must be simple, clear and to the point
can act alone, usually acts in a supporting role to a sovereign application.

postures for the web
- Web Applications (Sovereign Posture): google drive, google Docs, facebook, similar to desktop apps
- Transactional (Transient Posture): Navigation Structor (browse & search functions), blance between Sovereign and transient.
- Informational

Psoture for smartphones and tablets
- Smartphones: Transient - use more on the go, doing more simple tasks
- Tablets: Sovereign - you sit down to use it more longer

Chapter 10 - Optimizing for Intermediates

Beginner
- in a hurry
- need help in the beginning
- want to get an intermediate very quickly
- need a mental model to understand the scope & concept of the product
- need overview informations
- rely heavily on menus

Intermediates
- familiar with the interface
- need reminder for functionalities —they forgot, how to use.
- need fast access to functionalities
- don't need scope & purpose, they already know that

Export
- want shortcuts
- want high customization
- high information density
- want to work fast,
- don’t want to search or browse for functionalities

Design sovereign softwares - beginners, intermediates, experts

Beginners
- Guiding
- good wording in menues
- mental model to functionalities
- help functionalities

Intermediates
- fast access to functionalities
- Tooltips
- information about advanced functionality, that they believe in future use
- customization
- minimized navigation

Expert
- shortcuts
- customization

beginners --> intermediates?
- order functionalities in frequence of use
- introductions
- help functionalities
- mental models (understand - concept and scope of functionalities)
- tooltips

Chapter 11 - Orchestration and Flow

“harmonious interactions”
- mental model
- status & modeless feedback
- relevant tools & relevent Informations for relevant tasks
- less is more
- avoid irrelevant (discussion, asking questions, reporting)

Animations in a graphical interface
- Sense of physicality
- Shows sth. is happened
- focus attention
- Fedback

Chapter 12 - Eliminating Excise

Excise?
What you have to do and that is not goal-directed, small tasks that do not directly contribute to the user goals. (Inefficient and unnecessary)

3 types of excise
- Navigational Excise
  - navigation across multiple screens, views, or pages
  - Navigation between panes
  - Navigation between tools&menues
  - navigation of informations
- Stylistic Excise/ visual Excise
  - Skeuomorphism Excise
  - finding iteams in a list
  - where to begin to read
  - which is clickable
  - what is decoration
- Model Excise
  - Try to avoid unnecessary modal controls
  - error notification
  - configuration messages
  - don't make users ask for permission

eliminate excise?
- Reduce the numbers of places to go
- provide signposts
- provide overview
- do not replicate mechanical models
- avoid hierarchies

Mapping controls
Poor mapping requires the user to top & think about the relationship, break the flow.
Can result in user errors.

Chapter 13 - Metaphors, Idioms and Affordances

Affordance?
“How the thing could possible be used”

Pliancy
Pliancy & hinting, communicate to a user, how an interface element can be directly manipulated. It makes interactions visible to the user

4 types of pliancy hinting
- static hinting: a function or object in his default state but it draws attention to itself by using shadows, colors, etc.
- dynamic visual hinting (dynamic hinting): the element change appearance when cursor passing over it
- Plaint response hinting: The element changes appearance when it is actually interacted with.
- Cursor hinting: The cursor changes Appearance

Idiomatic
Idiomatic UI’s are not focusing on technical knowledge or intuition of functionalities. Idiomatic design support for interaction designer to develop idioms which, simple to learn, non-metaphorical in visual and behavior to accomplish goals and tasks

Chapter 15 - Preventing Errors

Modeless feedback?
- which is just there.
- which does not interrupt.
- does not require special actions

undo - made more powerful?
When the user know what he actually will or had “Undo”.

Chapter 16 - Designing for Different Needs

pedagogic, immediate, invisible commands

Pedagogic Commands:
- Example: Labeled Buttons
  - for beginners and slow
  - need a short and clear as possible description.

Immediate Commands:
- Example: Drag handles, real-time manipulation controls like sliders and knobs, pushbuttons, toolbar variants
  - for “perpetual Intermediates (Expert)”
  - visible but without text

Invisible Commands:
- Example: Keyboard accelerators, gestures (swipes, pinches, flicks of the finger)
  - for “perpetual Intermediates and Expert”
  - very fast
  - have to be memorized
  - working sets - Commands you already know

Accessibility in graphical interfaces
- follow OS accessibility standards
- don’t rely on just colors, use contrast/frames/etc.
- don’t override user-selected system settings
- provide only keyboard navigation
- avoid blinking, flickering and such things

Chapter 17 - Integrating Visual Design

Visual attributes - graphic components
- context
- Shape
- Size
- Color, RGB, HSV (Hue, saturation, value)
- Orientation
- Texture
- Position
- Text & Typography
- Information Architecture
- Motion & Change over time

benefits - grid
- It helps to succeed in usability, Aesthetic Appeal, Efficiency.

Through:
- Align controll elements like checkbox, radio button, etc. in a grid
- align across controlls groups & panes, everything should follow the same gid principles.
- a grid should have relationships between different sizes (Ratio)

Chapter 18 - Designing for the Desktop

“anatomy of a desktop application”,
- Primary primary window: covers the application/screen, splitted in: content Pane, index Pane (Navigation), other Panes
- Secondary Window: supporting first window with less frequently used properties & functions

Menues - “pedagogic vectors”
- Menus should represent all/ most frequently functionalities in the application.
- look through the menu he got a hint about the power of the application.
- Icons in menus for recognize functionalities without having to read.
- Accelerators (Short-keys) as idiom for faster use of fkt.
- Disable menu items in context.

Difference: toolbar & tool palette
- Toolbars: Icon + Textlabel / only Icon: perform actions on selections
- Tool Palettes: Switches between states

Cascading menus
- efficient way to screaming functionalities
- for big applications
- can be used in a right-click, fast access

Disadvantage:
- more mouse movements
- not a good idea for transient applications, better for sovereign applications

Tool tips
- clever & effective user interface idiom
- adds a pedagogical vector to an icon button without any of the drawbacks of text labeling.

object-verb oriented selection sequence
- You can select multiple objects before taking an action.
- You can restrict to only show the actions available for these objects.

Types of selection
- Selection:
  - Several function can be done in a row
  - after selection, only possible commands are shown
- Multiple Selection:
  - by hierarchy or category
  - Arbitrarily:additive selection — via modifier key (shift + mouse), dragging a marquee (select box)

drag and drop sequence
- only one source and one target
- can have multiple drop candidates - highlight these (targets)
- Drag cursor must visually identify the source object
- Insertion target - indication in a text
- The software have to be receive a visual feedback if it’s done.

Chapter 19 - Designing for Mobile

Touch interaction advantages
- direct manipulation
- allows for multi-touch (zoom, etc)

touch interaction disadvantages
- finger point is not preceise
- occlusion, if you point on the screen you occlude a part of the screen
- one-hand interaction, it reaches the whole screen
- No hover/ mouse over
- no dynamic visual hinting pliancy
- no tool-tips

Chapter 21 - Controls and Dialogs

Difference between imperative controls, selection controls, and entry controls?

imperative controls
- command actions
- response to a verb
- The verb - executes immediately

selection controls
- action required object

entry controls
- are bounded and unbounded, with validated entry control (active - notify rejection, give hints)

Imperative controls
- buttons
  - actions & functions
  - Execute by tap, click & release using the mouse
  - default buttons are highlighted, these are the most common controls
  - the most easy discoverable idiom
  - affordance is the visual pliancy, indicated its pressability
  - pointing, clicking/taping visual change (indicates that it has been chaned)
  - dynamic visual hinting
  - pliancy response

When use: used for actions
Why use: better visual hinting as a hyperlink

- Hyperlinks / Link:
  - images can also be used for links
  - direct & useful interaction idiom
  - typically with blue underlined text

When use: navigation through content
Why use: imperative control for navigation

Selection controls
- Check Box:
  - require well writte text
- Toggle button
  - two-states check box
- Drag & drop between list
  - object can be drag & drop from a list in another
- Dropdown list / combo Box
  - space efficient radio button
- Combo Icon Button
  - space efficient radio button, with more content
- Switches
  - common on mobile devices
- Barrel control
  - Controlling by swiping
  - best suited for mobile

Entry controls
- Unbounded: dials, slider, spinner
- Bounded: free text(input fields)