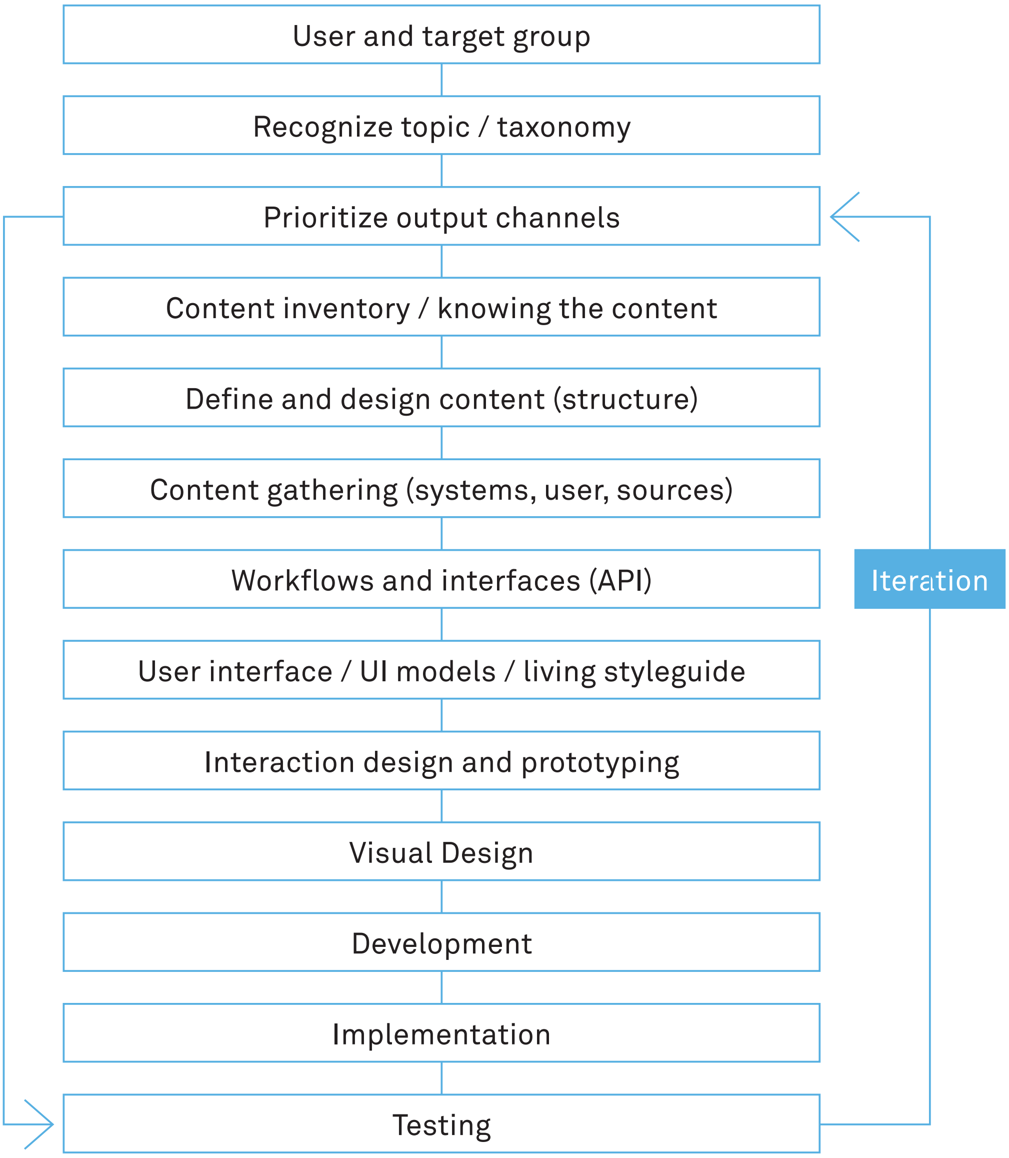
Multiscreen Project Workflow

A process recommendation that supports and delivers multiscreen-ready results

Addition to the »Multiscreen UX Design« book

The following diagram is just a rough overview. There are a lot of helpful methods, that have to be considered and applied (see following pages).

Start with the users and the topic. When you know these initial points you can successively get to know and work on the content. The user is always important, that means that authors, editors and recipients (all of them!) should always be taken into consideration. Don't forget to consider the potential context of use. Content modeling, UI modeling, visual design and development are iterative and overlapping project phases. Such a process is seldom linear, phases can parallel, change and swap. When you think you're done, think again (if possible). Define, create, refine, repeat....



User and target group

Recognize topic / taxonomy

Prioritize output channels

Content inventory / knowing the content

Define and design content (structure)

Content gathering (systems, user, sources)

Workflows and interfaces (API)

User interface / UI models / living styleguide

Interaction design and prototyping

Visual Design

Development

Implementation

Testing

Iteration

MULTISCREEN EXPERIENCE DESIGN

Multiscreen Project Checklist

A (incomplete) checklist for orientation in multiscreen projects.
Feel free to add or ignore (irrelevant) points.

Addition to the »Multiscreen UX Design« book

The checklist is not to be understood as a chronological list. It is separated into several disciplines within a UX/Multiscreen project (these areas can overlap in the timeflow (e.g. first considering the user, then content topics, afterwards user again, than UI questions and finally again content, before prototyping and development starts, etc.)

User and Target Group

- □ Analysis and definition of users and target group(s)
- □ User behavior (status quo and prediction and/or goal)
- □ Interpret existing statistics (multiscreen-relevant and -representative statistics)
- □ User prototypes / media user types / define and develop personas → see Chapter 3
- □ Polarity profiles / semantic differential / experience mapping
- □ Prioritize user and content → see Chapters 3 and 7
- □ Evaluate context of use → see Chapter 4
- □ Evaluate usage intention
- □ Work out Touchpoint Matrices → see Chapter 3, Section 3.2.1
- □ Generate Multiscreen Dayflows → see Chapter 3, Section 3.2.2
- □ Prioritize device classes / define key device → see Chapter 2, Chapter 5, Section 5.1
  (device classes / Which device what for? Device prototypes / Mobile First, Tablet First, etc...)
- □ Usage patterns and experience categorie → see Chapter 2, Section 2.6

UX Concept

(Generally / overall)

- □ Ideation (I) / objective: What do I want to offer to the user?
- □ Define UX vision in one sentence from the user‘s point of view.
  Guidline for orientation (e.g. to double-check ideas)
- □ Outline UX concept approach
- □ Evaluate possible/ interesting patterns and principles for the »usage of the service« → see Chapter 5
- □ Priority list(s) 1-10
  Each topic/idea needs to be prioritized / no identical priorities / discard 6 to 10.
  A priority list can basically be compiled for everything: user type, device type, viewports, target channels (channel type), content, concept/ideas, UI and Visual Design moods, patterns, etc.
- □ (Paper) prototyping... Afterwards/ based on ...: asking questions, gathering insights, generating ideas
- □ Ideation (II) / adjust objective and concept if necessary. Are the priorities still valid?

Content

□ Content analysis and definition → see Chapter 5, Section 5.10 and Chapter 7

□ Content inventory (kind and amount of content) → see Chapter 7, Section 7.2.2

□ Content prioritization / Find the one »hero content«

□ Check usefulness of the contents. Prioritize content from 1 to X per each device class.
Allocate each position only one time.

□ Content modeling / content types → see Chapter 7

□ Content reference wireframes
Order and allocate elements from the content inventory to the wireframe. The wireframe shall be responsive (viewport relevance / cf. preview-feature).

□ Content Reference Wireframe Iteration

□ Hierarchy and structure: Design in text (structured content)
Texteditor and HTML structure / build hierarchy with <h1> und <p> und <li>, etc. / HTML 1.0-ish
Note: The world's first website is almost mobile-ready
http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html)

□ Linear Design

□ Content analysis provides relevant UI characteristics (and therefore a predefinition for the UI modeling)
Consider: The content (model) defines the UI (Model) → see Chapter 7, Section 7.2.5

□ Content creation / Content Manager Experience Design → see Chapter 6, Sections 6.4. and 6.5

□ Define and setup »content management system« → see Chapter 6, Section 6.3, and Chapter 7, Section 7.3

UI / VD

□ UI Inventory → see Chapter 7, Section 7.2.2
(only sensible when there's an existing UI, e.g. with a relaunch or redesign)

□ UI Modeling / UI elements / Atomic Design → see Chapter 5, Section 5.8.1 and Chapter 7

□ Patterns: Responsive Design patterns, usage patterns, interaction patterns, navigation patterns, etc.

□ Coherent patterns (size, thumb, touch, mouse, etc.) / interactive states

□ Define channel type (unit / topic, cf. content types)

□ Define website type (adaptive, responsive / elastic, fluid, fixed)

□ Define layout type(s) → see Chapter 5, Section 5.8.1

□ Develop wireframes
As central documents that are updated during the whole process and have to offer a click-dummy at once.
It's possible that hybrid forms of website type and layout type arise during the process, even possible at module level.

□ Prioritization of device classes and viewport
(relevant for interaction modes and usage behavior)

□ Identify breakpoints
(Breakpoints graph: Where are the relevant breakpoints?)

□ Scribble, sketch, etc.
Design for various breakpoints: Start with the small screen first, expand until it does not look good anymore (then it's time for a breakpoint)

☐ Define responsive design patterns

☐ Develop layout raster

☐ Pattern library / dynamic raster

☐ Visual Design
(style tiles vs. photoshop mood screens vs. moodboards)

☐ Generate style-tiles or moodscreens based on UI components and elements

☐ Based on UI modeling and Visual Design: Living Styleguide → see Chapter 5, Section 5.8.1
(becomes the documentation / parallel to implementation)

Interaction

☐ Paper prototyping (low-fidelity) of core funcionalities
several variants and iterations (intersection from UI to interaction to prototyping to implementation)

☐ Elaboration of the concept (based on paper prototypes for example)

☐ Define navigation patterns

☐ Define interactions (and transitions) per viewport

☐ Check and verify Visual Design concerning interaction styleguides and guidelines of different operating systems

☐ Plan additional time to achieve maximal usability with core functionalities
Gamification, Micro UX, Transitions, Microjoyment → see Chapter 5, Section 5.16

Prototyping / realization

(building a digital high-fidelity prototype / in dialog with external service providers if necessary)

☐ Define and apply performance budget → see Chapter 5, Section 5.8.1

☐ Evaluate and choose apropriate frameworks / templates / technology
(not a specific UX topic, talk to and discuss with developers and technology experts)

☐ Components → devlop modules

☐ Viewports and breakpoints (Major and minor) with mediaqueries (based on content design)

☐ Plan production of media content for viewports/ devices classes (dimension and pixel density)

☐ Develope and produce interactive styleguide / pattern library / component library → see Chapter 5, Section 5.8.1

☐ Templates with the aid of the styleguide and component library

☐ Implement according to »progressive enhancement«

☐ Prototyping

☐ Prototype screenshots (for different breakpoints)

☐ Prototype > Finetuning > Completion (provided that realization belongs to the project)
target platform can also be conceptualized modularly (app, website, foreign, etc.)

☐ Testing

Testing / iteration

□ Introduce front end styleguide / pattern library

□ Define test cases and test process

□ Test with web developer tools in Firefox, Chrome, etc.

□ Test with simulators/ emulators (Xcode / Android SDK or Genymotion)

□ Test with a service like Browserstack

□ Test alle devices with Ghostlab in a shared WLAN (http://vanamco.com/ghostlab/)
Everyone sees the same on their device as the main tester

□ Use iOS Network Link Conditioner for reducing performance (or alternatively adjust via router)

□ Use Open Device Labs

□ External Testing service provider

Find more information in the book. → www.msxbook.com/en

Helpful und inspiring resources (that I also used for my checklist):

- Responsive Design Workflow *by Stephen Hay* → http://www.responsivedesignworkflow.com
- The Atomic Workflow *by Brad Frost* → http://atomicdesign.bradfrost.com/chapter-4/

Note:

This checklist is not complete (it hardly can't be). Feel free to add methods and topics or ignore (irrelevant) points. I'd be happy to hear from you if you have any recommendations. Thanks!