

Wonderware design guidelines

1 Focus on real users
Identify and define

2 Identify real tasks
Prioritize

3 Collect real content
Be specific and constructive

4 Build a device strategy
Think pixels

5 Build the hierarchy
Structure

6 Don't forget the interactions
Build a qualitative user experience

7 Make it visual
Prioritize and be consistent

8 Remember the design process
Collaborate, learn, think positive

1 Focus on real users

Identify and define

« Obsess over customers : when given the choice between obsessing over competitors or customers, always obsess over customers. Start with customers and work backward. »

Jeff Bezos, founder and CEO of Amazon

User profile

Meet the end user

Don't ignore the end users, they are the key to a good interface. Observe and talk to them to understand their skills and experience.

Create personas


If you can't meet up with end users, don't think you can avoid it and design the interface for yourself. Build fictional personas with habits, skills and needs. Make them real.

Scenario

Don't start designing before you know exactly what your HMI is for. To know so, imagine a scenario structured by the different tasks your users would do in a day.

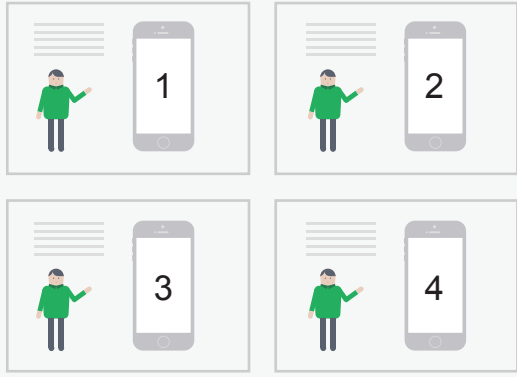
Co-create or share the story with the project stakeholders to build a shared vision.

USER PROFILE



Name :	Goals :
Age :	Main tasks :
Occupation :	Rare tasks :
Hobbies :	Obligatory tasks :
	Problems :
Devices :	Operating procedures :
Favorite applications :	Wishes :

SCENARIO - day 1



1 2 3 4

Reminder

Here are some questions which will help you imagine the needs of your users :

Profile

- What is their name ?
- How old are they ?
- What is their job and salary ?
- Are they in a relationship ?
- What do they like to do in their free time ?

Digital literacy

- What devices do they own ?
- What devices do they use at work ?
- What are their favorite applications ?

Work habits

- What is their goals during the day ?
- What do they do most often ?
- What do they do seldomly ?
- How do they do it ?
- What do they do when nothing is happening ?
- What problems can occur ?
- What do they do when something bad is going on ?
- What do they feel they need and wish ?

2 Identify real tasks

Prioritize

Task inventory

Don't ease into the computer too quickly, work with a pen and paper or use a whiteboard to isolate your user's key tasks.

Make an inventory of all the tasks in the scenario and list all the situations that have arisen (servicing, restarting, change of procedure...).

Functional categories

Categories

Classify the tasks into functional groups according to how often they are used and how similar they are : alarms, manual commands, settings...

Sub categories

List sub-categories which exist within the categories.

There shouldn't be more than 3 levels of depth of information.

Naming

Name the groups of functions with words familiar to the user. They will help build your navigation elements.

TASKS

- Tasks in interface**
 - Adjust set points
 - Launch procedure
 - Fill reports
 - Respond to an alarm
- Tasks on site**
 - Check for misplaced product
 - Refill material
 - Replace die
- Situations**
 - Beginning of shift
 - Setting up
 - Production
 - Changing order
 - Cleaning in place
 - Stopping the machine
 - Emergency shut down
 - End of shift

CATEGORIES

- New shift**
 - Log in
 - Check performances
- Changing order**
 - Select new workorder
 - Replace die
 - Adjust set points
- Production**
 - Adjust set point
 - Fill reports
 - Respond to an alarm
 - Check for misplaced product
 - Refill material
 - Replace die

Reminder

Put the most important groups first and remove all redundant or irrelevant information.

The main information should be accessible in maximum 3 clicks or taps.

NAMING

DON'T

TIC00_X22.27°
In section 22 of machine XXFGH the average temperature measured in Celsius is about 27, 265875°.

✗ Don't use too much abbreviations and cryptic text

DO

Dryer / Section 22
Temperature : 27 °

✓ Be explicit but concise.

3 Collect real content

Be specific and constructive

Data

Don't ignore complexity. Know how detailed the interface must be and write down the units that need to be represented. The values shouldn't be too long, if they are it means the unit should change (mm to cm, g to kg...). Always display the relevant unit of measurement to avoid input errors. Understand the relationships between datas and actions and find the best objects to represent them (tables, pie charts, diagrams...).

Editorial content

Clarity

Don't communicate in a lengthy and boring manner. Write in the users own language. When possible, use full words and sentences, as opposed to abbreviations and codes. Try to accompany visual icons with text.

Consistency

Create consistency in the syntax and tone (neutral or directly addressed to the user...).

Build a positive discourse

Your messages should be worded positively and affirmatively and provide constructive insight into how to use the application properly.

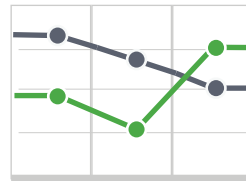
DATA REPRESENTATION



Repartition



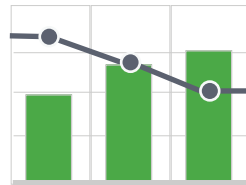
Percentage



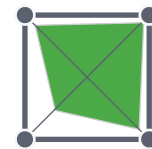
Trends



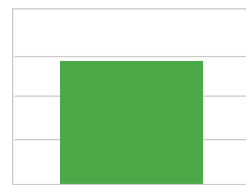
Balance



Compared trend



Relationships



Level



Load

Reminder

To build a positive discourse, for instance :

Don't say :

"You have entered the wrong information"

But try :

"An account number should be eight digits in length."

DIALOG COPY

DON'T

Do you wish to cancel the set point change ?

OK

Cancel

DO

The new set point is 25 liters. Do you wish to save it ?

No

Yes

The copy must be clear, concise and consistent with the buttons to avoid confusion.

4 Build a device strategy

Think pixels

Soft and hardware hand in hand

Don't think an interface is always the same, it isn't. It is build and conceived according to the device it is going to be used in.

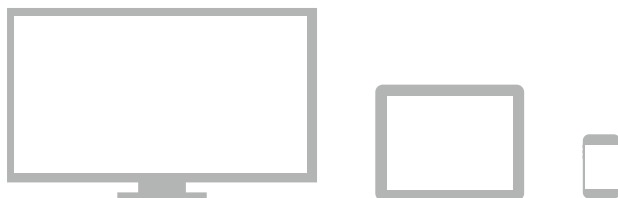
According to the user's needs, the market and your technical possibilities, determine the devices that your interface will be used in.

Grid

Once you decide on a device strategy, you know how much available space you will have so build a grid for all you elements accordingly : 1280/1024 px tactile, 1024/930 px...

Standards

If possible, use the navigation and interaction standards of the device you are using to minimize the cognitive load of the user.



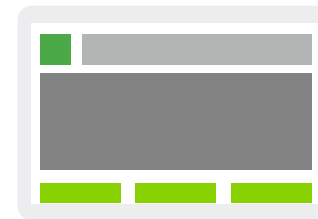
SCREEN SIZES



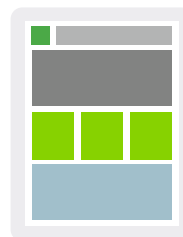
1920/1080 px



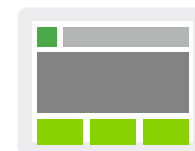
1440/900 px



1280/1024 px



768/1024 px



800/600 px



320/480 px



HD : 640/1136 px
Perceived : 320/568



HD : 1080/1920 px
Perceived : 540/960 px

Reminder

Here are some questions which will help you decide on the device(s) :

- Is it an old PC ?
- Is it a tactile screen ?
- A phone ? Android, Apple, Windows 8 ?
- Is it present on multiple devices ?
- Is it different according to the device or is it responsive ?
- Is it a software, a web site or an application ?

TOUCH SIZES



26/26
px

Minimum touch size



7 mm



34/34
px

Average touch size



9 mm



8 px

Minimum spacing



2 mm

5 Build the hierarchy

Structure

Navigation

After making functional clusters, build a navigation bar that allows navigation at all times between the main pages of the interface (commands, alarms...).

Clusters

Items that are logically connected should be grouped together on the screen, whereas items that have nothing to do with each other should be separated. You can use white space between collections of items to group them and/or you can put boxes around them to get the same result. Generic functions such as help, connection, language can be assembled. They can be placed in the upper bar with the logo, status, time and alarms...

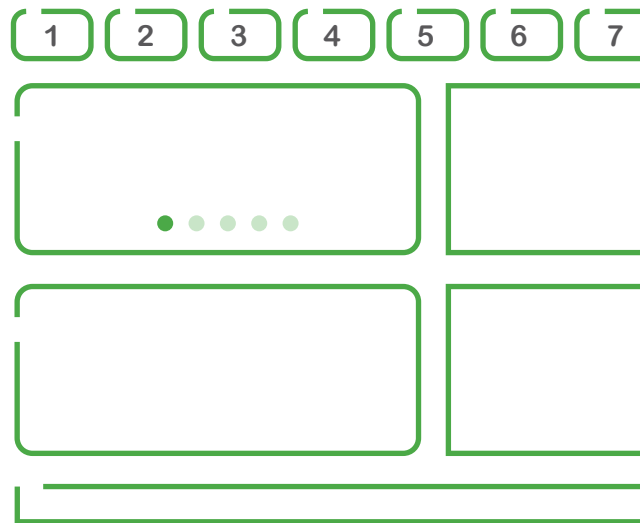
Readability

Define a readability dynamic and place elements accordingly.

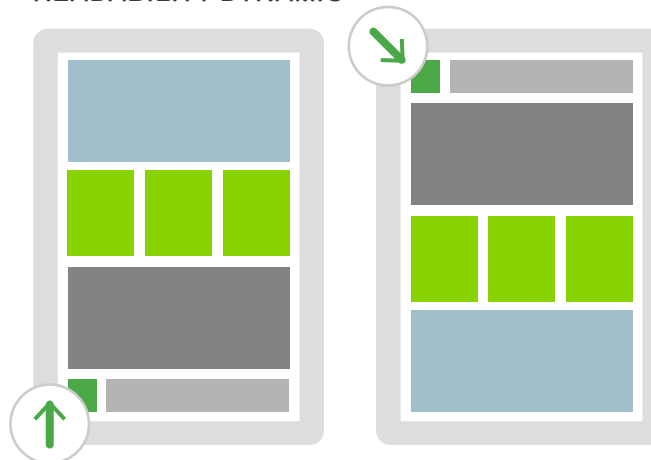
Ex : top left to bottom right following the western societies reading dynamic or down to top following the Windows users habits...

NAVIGATION

Maximum 7 items in a navigation bar



READABILITY DYNAMIC

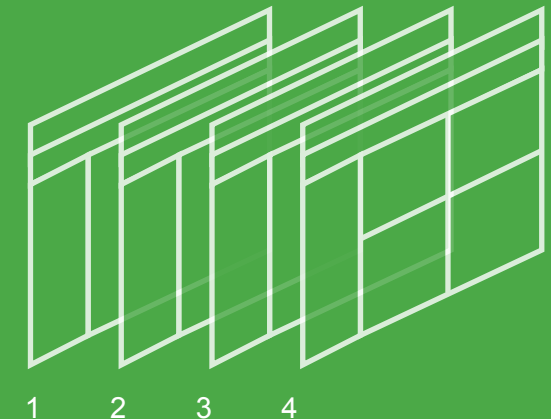


Reminder

At all times the users must be able to :

- Situate themselves clearly
- Access the main navigation functions
- Switch function or task
- Move to the previous or next step
- Log in or log out.

Try to keep the same layout from one page to the other.



6 Don't forget the interactions

Build a qualitative user experience

Feedbacks

Always inform your users with visual cues or messaging of actions, changes in state, errors and exceptions.

Provide different types of notification according to the severity of the message.

Always include the *ok* or *close* function to start a process, confirm an order or after entering data.

Make way for errors

Every action of your user should be undoable. If the user does cause an error, use your messaging as a teachable situation by showing what action was wrong, and ensure that they know how to prevent the error from occurring again.

Animation

Don't disturb your users for nothing, avoid flashing animations unless they express a fatal emergency.

Things that are not the same shouldn't look the same.

Differentiate the static information from text fields or other dynamic elements.

The best interface designs are invisible.

FEEDBACKS

The image shows four examples of feedback messages in a light gray box with a colored header bar. 1. Information: Blue header with an 'i' icon, text 'The recipe's new parameters have been saved.', and an 'OK' button. 2. Validation: Green header with a checkmark icon, text 'Do you want to save the new recipe?', and 'NO' and 'YES' buttons. 3. Notification: Yellow header with a warning icon, text 'The packager roll will be finished in 1 hour.', and an 'OK' button. 4. Alarm: Red header with an alarm icon, text 'Please check the filter in the main dryer.', and an 'OK' button.

BUTTON STATES

The image shows three 'OK' buttons in different states. 1. Active: A white button with a gray border and the text 'OK'. 2. Selected: A white button with a green border and the text 'OK'. 3. Disactivated: A gray button with a gray border and the text 'OK'.

INPUT TEXT

The image shows two examples of input text. 1. 'DON'T': A red header with the text 'DON'T' and a text input field with the placeholder text 'Enter Colour'. 2. 'DO': A green header with the text 'DO' and a dropdown menu with the placeholder text 'Select Colour' and three options: 'Pink', 'Orange', and 'Green'.

✗ Free text increases the chance of input error

✓ Default values reduce the chance for error

Reminder

Consistency

When an application works consistently, it means that the users only have to learn the rules once, increasing their efficiency at each interaction.

Create patterns by applying similar codes and actions to your design and if possible replicate patterns existing in interfaces popular to the user.

Put your buttons in consistent places on all your windows, use the same wording in labels and messages, and use a consistent color scheme throughout your interface.

Velocity

The interface shouldn't be too crowded and heavy to avoid loading problems.

All interactions should be extra fluid, your users won't wait.

7 Make it visual

Prioritize and be consistent

Layout

Work on a fixed grid, keep the same page layout from one page to another.

With tactile interfaces, use as much screen space as possible, make elements big enough. Restrict the number of active zones and place them far enough to facilitate touch navigation.

Text

Only display essential information. The rest of the information can be displayed using filters, toggle buttons, pop-ins...

Make sure it is big enough to be read : minimum 14pt.

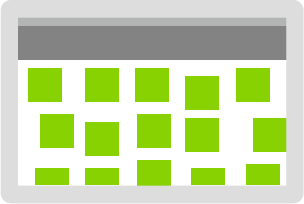
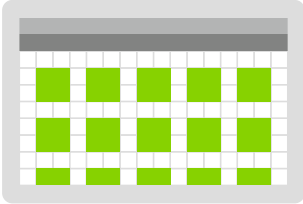
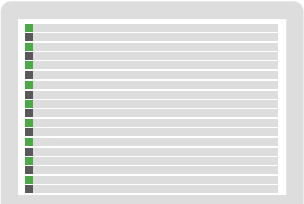
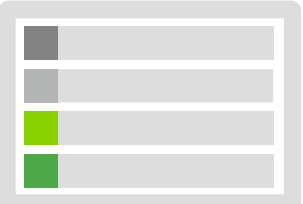


Test maximum character height and width : **A** is the tallest, **y** the lowest and **W** the widest.

Align fields : left-justify edit fields, the corresponding labels should be right-justified and placed immediately beside the field.

Justify data : right-justify integers, decimal align floating-point numbers, and left-justify strings

Icons

Limit the use of icons to standard functions or explain them with text.

DON'T	DO
 <p>✗ Don't design without a grid</p>	 <p>✓ Align all elements</p>
 <p>✗ Too small for reading and touch</p>	 <p>✓ Big enough for reading and touch</p>
<p style="text-align: center;"><u>Action</u></p> <p>✗ Action is different from a link</p>	<p style="text-align: center;">Action</p> <p>✓ A button is for an action</p>
<p style="text-align: center;">A link</p> <p>✗ Plain text is different from a link</p>	<p style="text-align: center;"><u>A link</u></p> <p>✓ Underlined text often say link</p>
 <p>✗ Don't align with the ascenders</p>	 <p>✓ Align with the text X-height (not ascenders or descenders)</p>

Reminder

Create a library of standard elements :

- buttons
- dialogs
- grids
- lists
- menus
- sub header
- switches
- tabs
- text fields
- tooltips...

Color and images

Limit the number of colors to 5 or 6 and use them to create meaning. For instance, use green when all is ok, yellow when something could be changed, red when things go wrong.

Provide appropriate contrast, it has to be maximum 70 % : use dark text on light backgrounds and light text on dark backgrounds.

The resolution of the imported images must be at least equal to the screen resolution.

8 Remember the design process

Collaborate, learn, think positive

Iterate

When developing interfaces, it is often said that you need to fail fast and iterate often. You will make mistakes but it will get you further than you think.

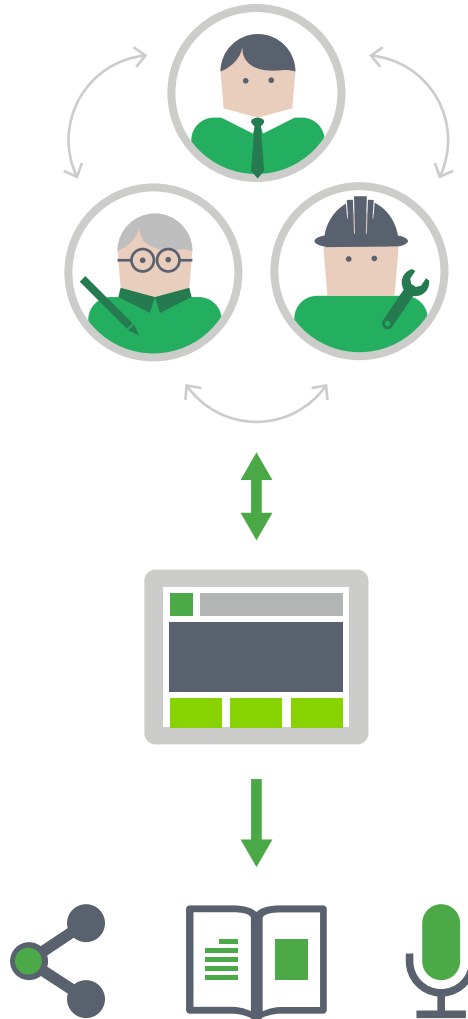
Co-create and share

Identify which stakeholders will be relevant for each phase of the project and work together with them. If possible don't just talk on the phone, meet up. You'll start to learn about each other's disciplines, reducing pain points and allowing more autonomy in the decision making.

Show your sketches around to stakeholders, managers, engineers, designers and users. Take note of the reactions, explain your decisions, discuss and if necessary come to a compromise. Always move forward.

Communicate

Communicate on your innovation in conventions, publications, social medias or other appropriate communication channels.



Reminder

Talk with your stakeholders to establish the tools that will be used and the way you will work together. Ask yourselves :

- Will the project be waterfall or agile ?
- How often will there be deliveries ?
- How will the files be sent, on what platform ?
- What will be the ways to keep track of changes and requests ?