ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

User Experience Design part II

Fabio Vitali

Two user-oriented process models

A task-oriented model

- *ISO 9241-210 (2010)*
  - *Official international standard, originally from UK*
  - *Aim: usabilty design*
  - *Five phases: Feasibility study, User Requirements, Implementation, Evaluation, Deploy*

A goal-oriented model

- Jesse James Garrett (2011)
  - Well-known professional, USA, word-of-mouth (passaparola)
  - Aim: User Experience Design
  - Five planes: Strategy, Purpose, Structure, Skeleton, Surface

The Elements of the User Experience

By Jesse James Garrett

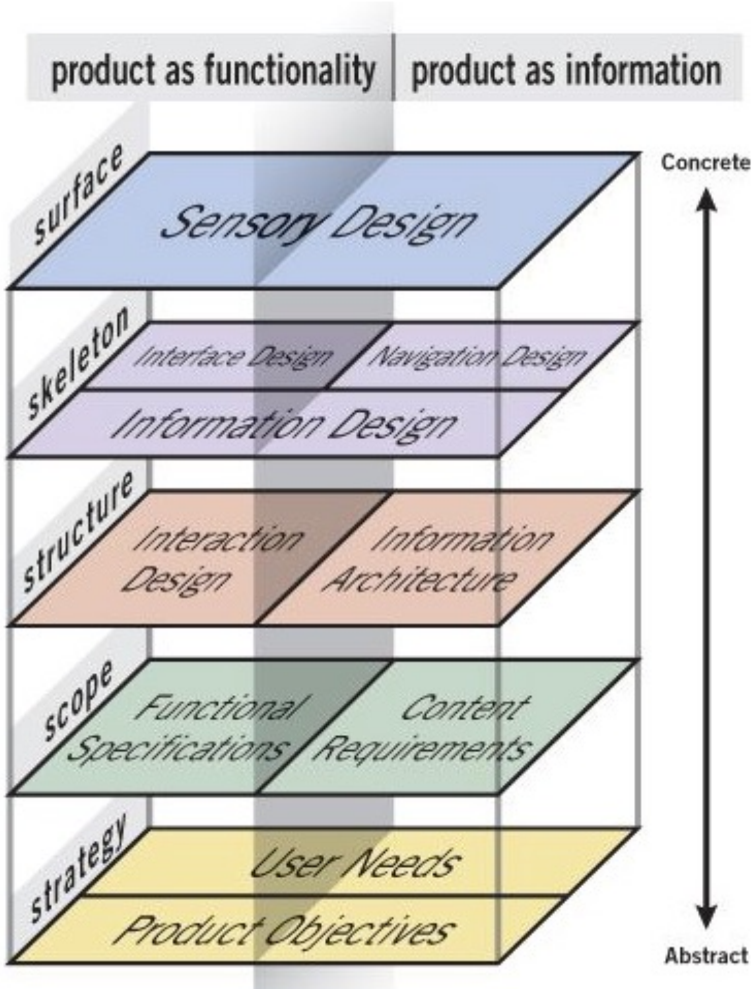
First a crudely drawn schema passed around by word of mouth by web designers.

Later a web page, finally a book (2006 and then 2010) providing a conceptual model and a series of implementable steps for managing a User Experience Design Process.

Jesse James Garrett invented the term AJAX in 2005.

Here I am extending Garrett's model including a number of other compatible ideas and approaches.

Garrett's schema

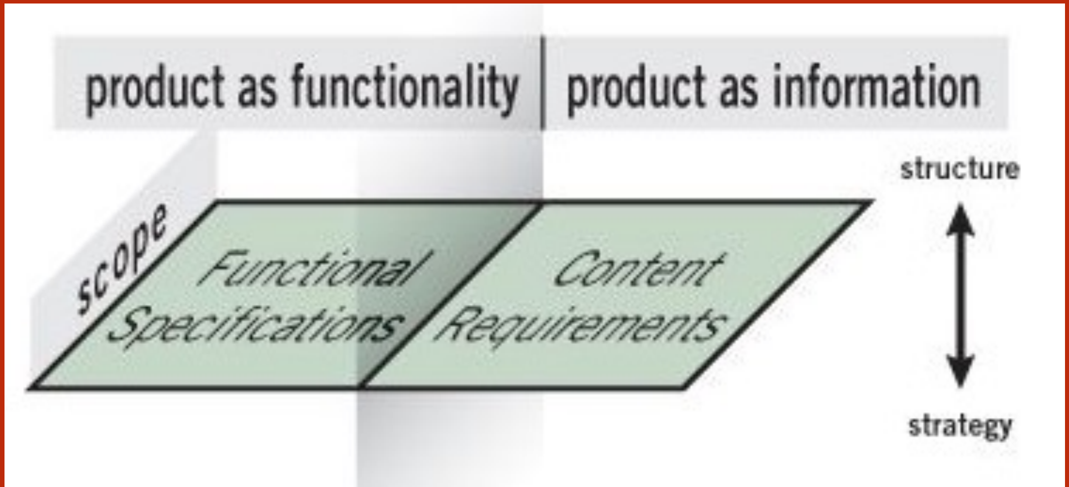


A linear process

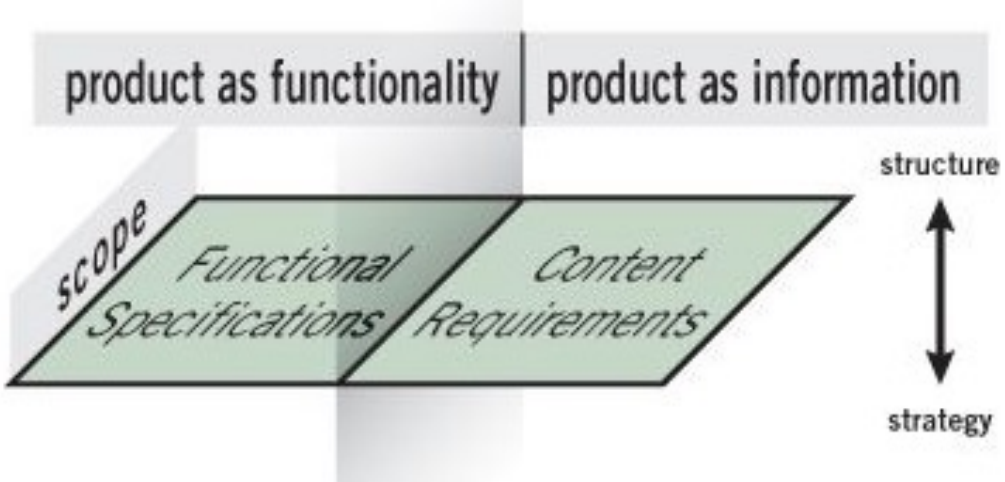
- from abstract to concrete
- mainly Web
- Parallelism between application sites and information sites
- It involves roles from management, architects, implementers, graphics, and sales.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The Scope plane



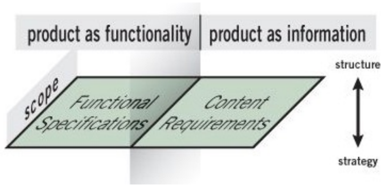
The scope plane



What are we doing?

What are we NOT doing?

- Functional requirements
- Content Requirements
- Scenarios

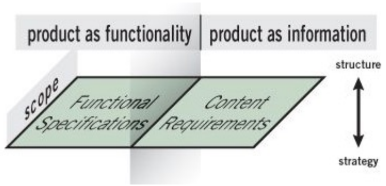


Functional & content requirements

Functional requirements or *functional specifications*. In some organizations they are two different documents:

- Functional requirements: at the beginning of the project, describes what the system should do,
- Functional specifications: at the end of the project, describes what the system actually does.

Content requirements: what information needs to be included in the content being developed.



Functional & content requirements

Writing requirements and specifications that work

- Writing the spec should not be a project in its own.
- Volume or detail don't solve problems, clarity and accuracy do.
- Specs don’t have to address every aspect, just the ones that arise confusion in the following part of the process.

Be positive

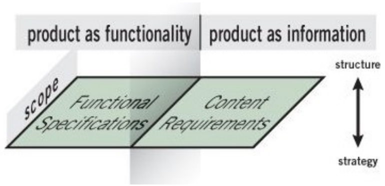
- No: The system will not allow users to buy a kite without string.
- Better: The system will direct the user to the kite string page if the user tries to buy a kite without string.

Be specific

- No: The most popular videos will be highlighted.
- Better: Videos with the most views in the last week will appear at the top of the list.

Avoid subjective language

- No: The site will have a hip, flashy style.
- Better: The look of the site will conform to the company branding guidelines document.

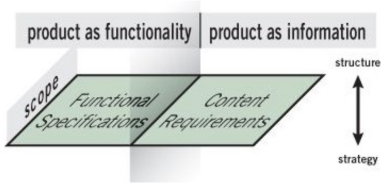


Scenarios

Storylines telling in detail how auser accomplishes a personal goal by carrying out one or more of the tasks planned on the system.

- Goal and task relationship
- Decomposition of user tasks in actions (internal and external)
- Identification of user's and system operations
- Narrative of user actions, its goals and its motivations to use the system
- NO specification of which features were used (the system as a black box)
- Establishes timing estimates and success criteria for the scenario and for each task within it.
- Scenarios are useful *both* the common situations *or* critical situations

Not only typical features, but also to stress test the peculiar characteristics of the system.



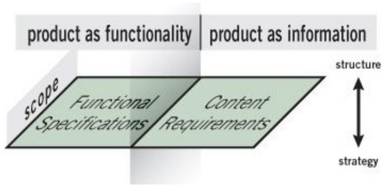
Design fiction

“Design fiction is the deliberate use of **diegetic prototypes** to suspend disbelief about change"

(Sterling, 2012)

“Design fiction is a mix of science fact, design and science fiction … The conclusion to the designed fiction are **objects with stories**. These are stories that speculate about new, different, distinctive social practices that assemble around and through these objects."

(Bleecker, 2009)



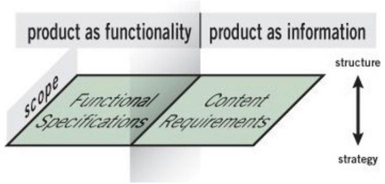
Diegetic prototypes

A diegetic prorotype is the product being designed immersed and functional to the narrative of the story. It is not the subject of the story, but a instrument in the development of the story.

The prototype is described as if it is a normal element in the life of the characters of the story, not a recent and wonderous innovation.

“The performative aspects of prototypes are especially evident in diegetic prototypes, because a film’s narrative structure **contextualizes technologies** within the social sphere. Narratives in popular cinema require certainty from their technological devices to **move their stories forward**."

(*Kirby*, 2010)



product as functionality | product as information

structure

strategy

scope

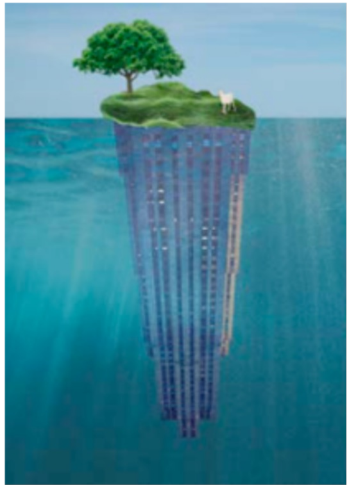
Functional Specifications

Content Requirements

Design fiction

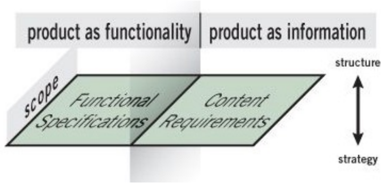
Poetics of Design Fiction

- What-if scenarios
- Basic Rules of fiction
- Design tools
- Diegetic prototype



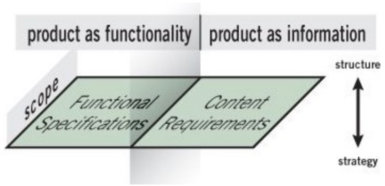
Pastiche scenarios

- Characters from movies or books
- Strong character-based scenarios



Phases of task analysis

- Identify the task to be performed
- breakdown tasks in few sub-tasks in terms of goals and sub-goals
- Write the task diagrams and check completeness (handling sub-cases, handling error, etc.)
- Check for a homogeneous breakdown of all details of each sub-task



A traditional example of scenarios

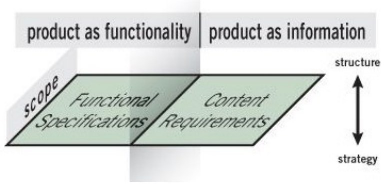
Help line for an online ordering system

A busy morning with a long queue of calls to the customer care office. Andrea has been working in this office for just a week and receives a phone call from Mr. Rossi.

Mr. Rossi has not received some merchandies he ordered three weeks ago. He provides his name and address. Andrea recovers his order, and controls it.

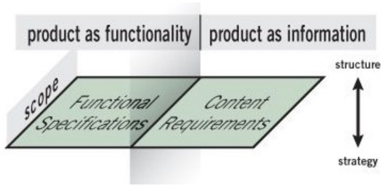
The merchandise was delivered to the courier a week ago, so Andrea gives Mr Rossi the tracking code of the parcel for further enquiries with the courier itself.

While he's checking Mr. Rossi's data, Andrea notes an error in the postal code, verifies it with Mr. Rossi and corrects it immediately.



Details of the example

- It describes tasks, not commands of the interface
- It is rather specific
- It describes a complete task
- It describes the users

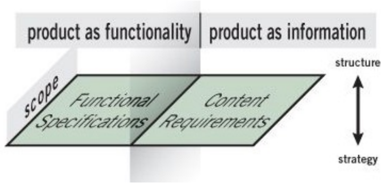


Task, not commands

No assumption is made on the types of command to be activated or the structure of the interface.

This genericity in spec can be used to compare equally design alternatives.

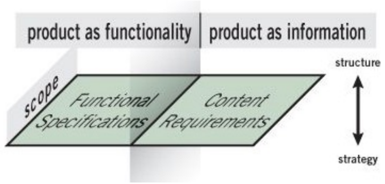
If we specified something like "Andrea types the name in the input field" we would have pre-described the "correct" way to accomplish this task, thus preventing the exploration of alternatives.



Extreme specificity

- It does not just say what the user needs to do with accuracy, but specifyes exactly which interface paths are affected.
- In practice, it allows (forces) to specify all kinds of detail that may sooner or later become relevant in evaluating design alternatives.
- In this example, for instance, it shows that it will be useful to present internal information (such as parcel tracking code) so we find it useful to show this information to the help center.

The case of long file names.



Complete tasks

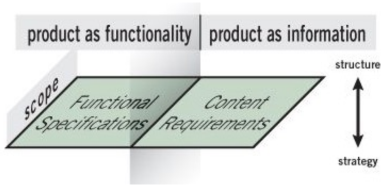
The description of the task is very specific and describes a complete situation.

This allows us to evaluate how well different aspects of the interface will work together.

The traditional software engineering requirements list is just a list of individual actions that the system must be able to execute.

It is not usually discussed how these individual actions are composed to accomplishing meaningful and complete tasks.

Start from managing input and output!

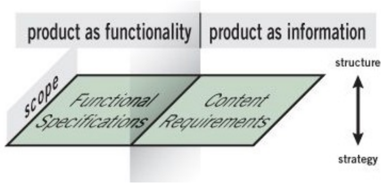


Exact description of users

Just like we detail the tasks exactly, similarly should users be detailed. This is the only way we know to immedesimate into the attitude, the psychology and the experiences of the user.

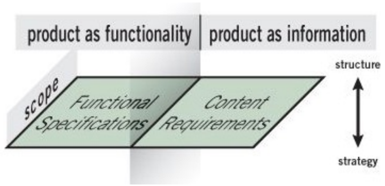
For instance, a medical expert system targeting doctors (who have no time to learn new tools) had an interface identical to the paper modules already filled in by hand by doctors in real life.

This type of decision would not have been taken if it had been decided to make an interface for medical technical assistants (who have been using computers for a long time and can be forced to learn a new tool).



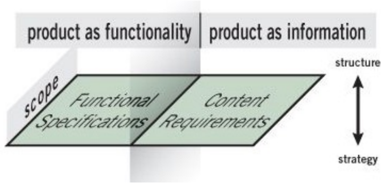
Some reflections on tasks

- The importance of the first impression
- Proportionate commitment
- Cue-routine-reward
- Habit loop
- Cognitive dissonance
- The flow
- Gamification



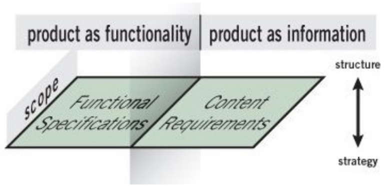
First impression

- You never have a second chance to make a good first impression.
- Our first impression is not necessarily obtained by using the system: we can watch a friend using it, or a guy in the seat next to me on a train.
- What kind of feeling do I get? What kind of impact does it have on me? Rewarding? Funny? Interesting? Useful?



Proportionate committment

- No user will dedicate all his efforts to make a system work.
- His/her dedication will be proportionate to the usefulness he expects to receive from the system
- Therefore, software tools that require more commitment than the expected usefulness of the result will not be used regardless of any other justification.



Cue-routine-reward

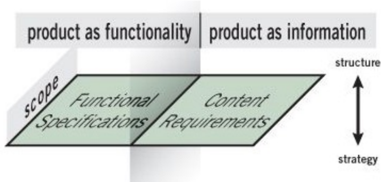
Reduces cognitive requirements of people trying to solve normal problems and varry out daily activities.

The trainer throws a stick (cue), the dog runs to fetch it and returns it (routine) receiving a cookie or a hug (reward).

- The cue is the even starting the reaction
- The routine is the frequent action connected to the cue
- The reward if the consequential benefit

Activities becoming routine are the simplest to perform and require no effort.

E.g.: mental activity of rats that have learnt the labyrinth is lesser than rats who are still learning it.



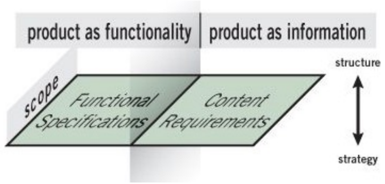
Habit Loop

The completion of an action, or the moderate success of a choice, will in our mind reinforce such choice or such action more than others potentially just as positive or even more.

After some repetitions, this action becomes a habit, never more questioned, and this choice becomes automatic and immediate. This *becomes a habit loop*.

Habit loops are dangerous when they lead to unbalanced behavior (eating disorders, lack of physical activity, bad attitudes, etc.). They are useful in general because they simplify our daily life by reducing the number of decisions and reflections that we need to make.

The habit loop is the result of receiving a reward after a routine. It generates an idea of proportionality in the commitment spent on our activities.

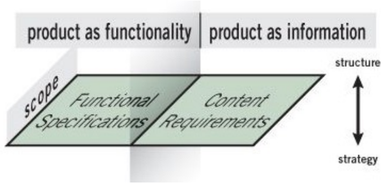


Cognitive dissonance and learned helplessness

Psychological conflict due to incongruous and simultaneous beliefs

If humans can not find a solution to a problem at a reasonable time, they consider themselves satisfied with a suboptimal solution even if we know that, somewhere and somehow, it is possible to find a better solution.

Dissatisfaction creates cognitive dissonance that lowers the expectation threshold



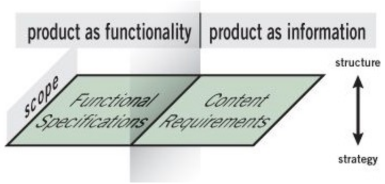
The flow

In sailboats, there is a magical moment where speed is enough to lift the boat over its own wake (scia), touching just the top of the water and reaching high speeds.

It happens suddenly and is a wonderful feeling. However, it is also a very fragile moment, as it takes just a clumsy maneuver to get back into the water and crash as if we hit a wall.

Humans have similarly a psychological state called "flow", which is suddenly activated when we focus on a task.

The "flow" is defined as a "profound and almost meditative involvement" on the task to be accomplished, and often induces a "gentle feeling of euphoria" and a loss of sense of time. In a state of flux, people are very productive, especially for creative or designing activities.



Flow and orchestration (1)

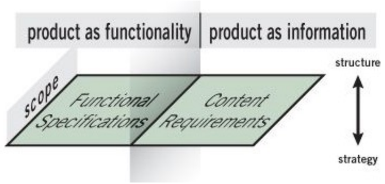
Like sailboats, for humans too, the flow is a magical and fragile state.

It is necessary to prevent that the clumsiness of the interaction interrupts it.

Once out of the flow, it is difficult and slow to get back.

To avoid getting out of the flow, these are useful ways:

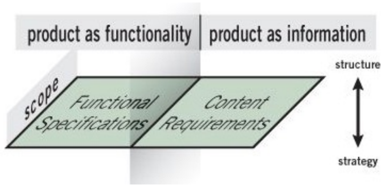
- **Follow mental models:** a tool that organizes its procedures around the user's mental model does not disturb.
- **Direct, don't discuss:** a steering wheel does not discuss: it limits to two directions the user's choice, but does not start a conversation with the user about the optimal choice of the next direction



Flow and orchestration (2)

- **Keep tools at hand:** a toolbar allows users to keep close the tools they need, and choose them quickly and easily.
- **Provide modeless feedback:** the easiest way to provide the user a feedback value is to show a modal window, but this requires the user to explicitly dismiss it. Non-modal information is a much better way, that can be shown to the interested user without bothering those who are not.

To achieve better interaction, the orchestration of the various parts of the interface into a single consistent and effective is the fundamental mechanism. The ultimate goal is the invisibility of the interface.



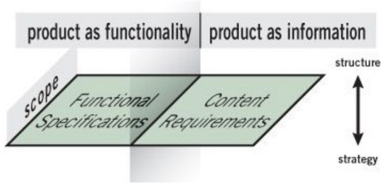
Gamification

Another important approach is the exploitation of our interest in the game challenge.

The *gamification* is the discipline that seeks to improve our perception by transforming routine activities into games.

They can naturally stimulate a flow effect and let us focus our attention on the task, so as to achieve better results with little effort.

The example of the operators of of X-ray scanners in airports



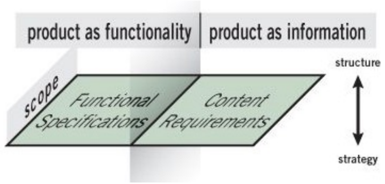
Paidia & Ludus

Paidia

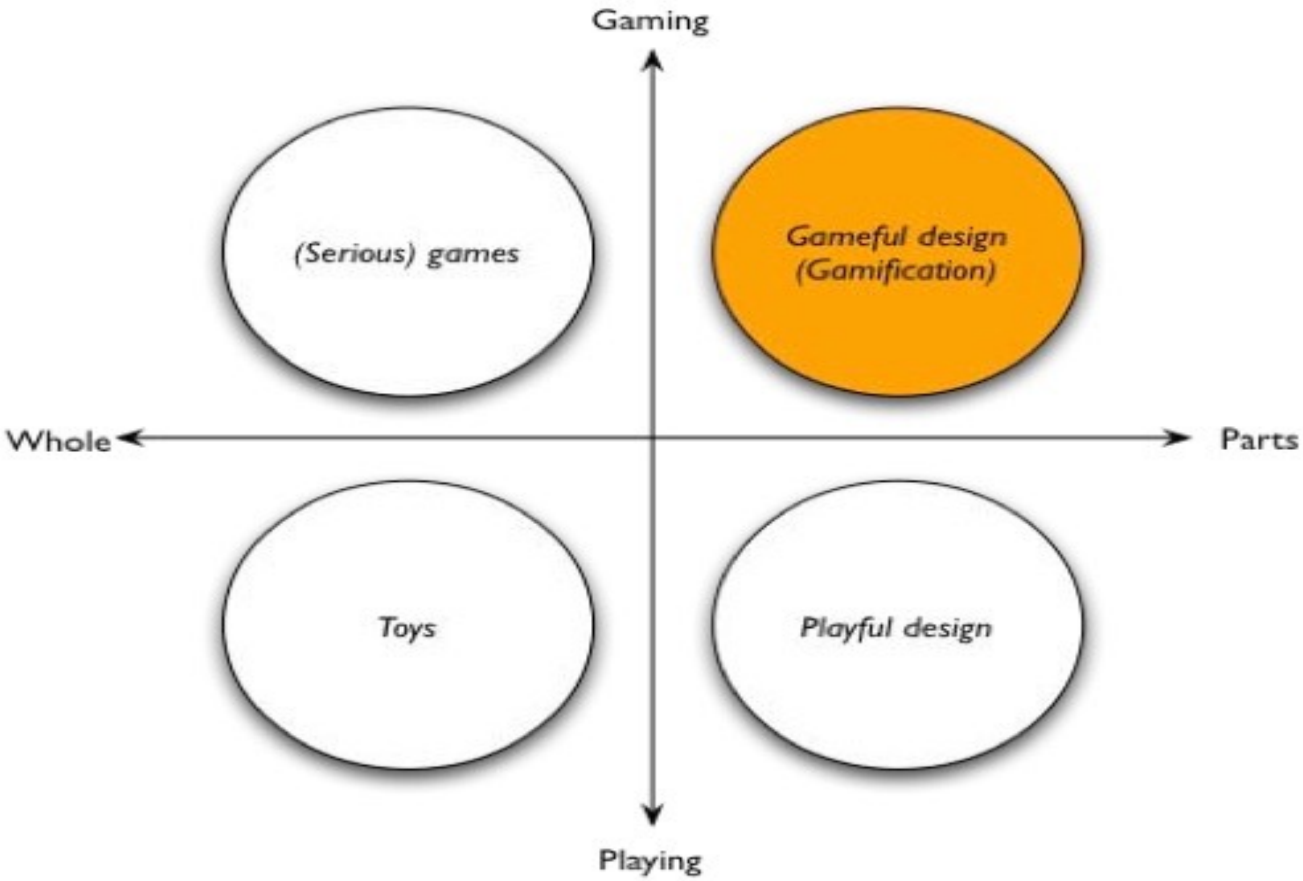
Unstructured & spontaneous activities

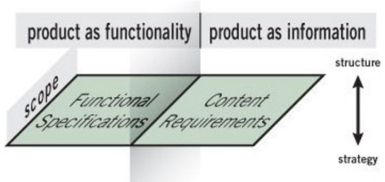
Ludus

Rule-based structured activities

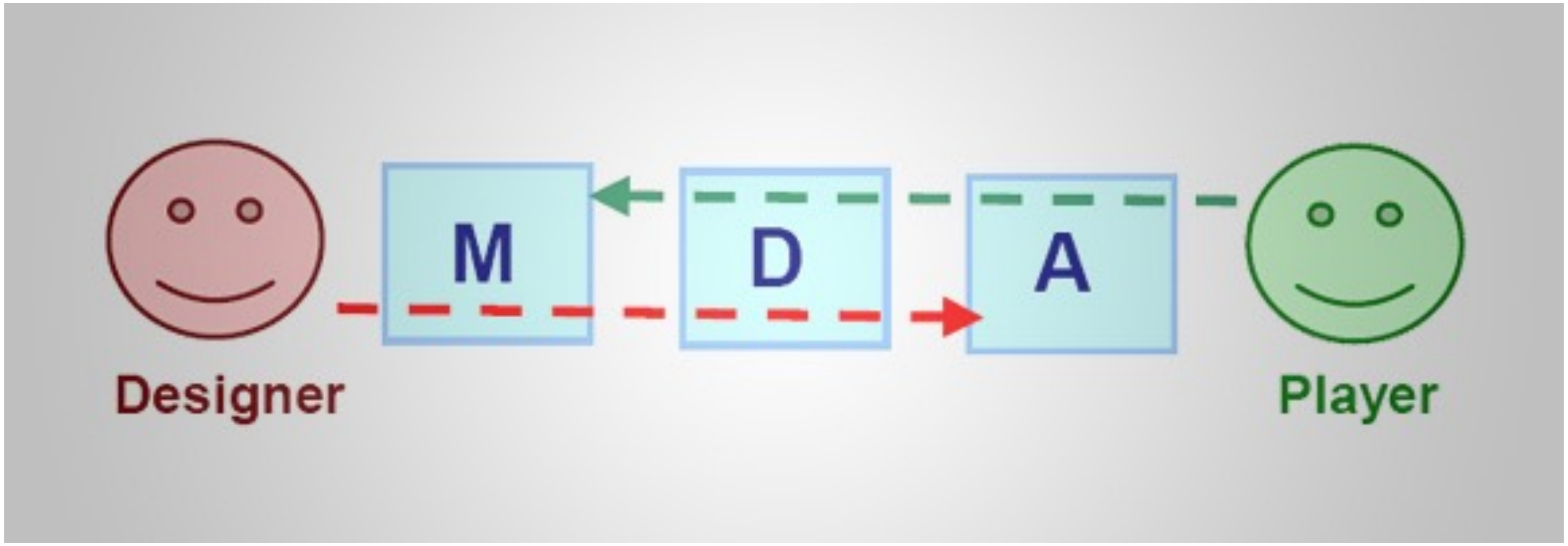


Gamification

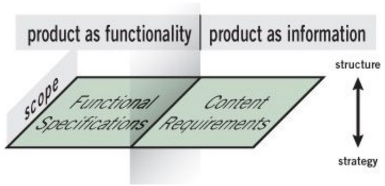




Components of gaming



The so-called Mechanics-Dynamics-Aesthetics (MDA) framework identifies three components in games: Mechanics, Dynamics and Aesthetics.

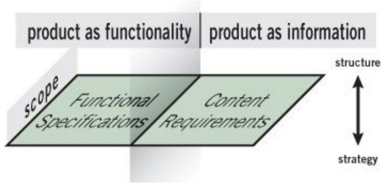


Mechanics

The mechanics are the atomic components of the game.

They are like the individual gears that make the game run smoothly and correctly.

They include game rules, number of players, roles of the players, the sequence of actions allowable, etc.

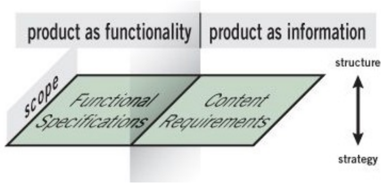


Dynamics

The dynamics describe the behavior of the game while the mechanics are being used correctly.

They include strategies, objectives (shared and private, long-term and short-term, etc.) and run-time events of the game

Trying to conquer Oceania in Risiko, collect swamps instead of forests in Magic, choosing black or white in chess are examples of dynamics in action.

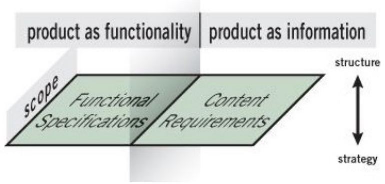


Aesthetics

The aesthetics represent the ways in which the game manages to entertain the players.

Mechanics and dynamics cooperate to evoke pleasurable sensations in players according to one or more emotions:

- Sensation: Player experiences unfamiliar feelings.
- Fantasy: Player watches or creates an imaginary world.
- Narrative: Player watches or creates a story that drives him/her back
- Challenge: Player urges to master some physical or mental ability.
- Fellowship: Player longes to be part of a community to be an active part of
- Discovery: Player wants to explore game world.
- Expression: Player exercises his/her own creativity.



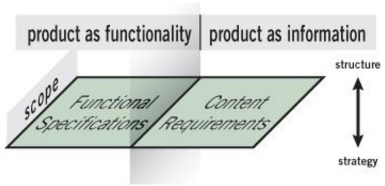
Motivations

Motivations drive us to carry out specific actions

We consider them as the set of the factors that drive an individual to exhibit a specific behavior.

Ignoring the motivations behing behaviors means failing the analysis.

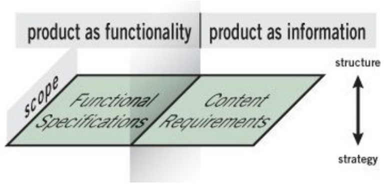
- Extrinsic motivations
  - External to the individual.
  - When we behave in some way in order to receive a reward or to avoid a punishment.
  - In gamification, extrinsic motivations are expressed by scores, merit badges, the desire to acend in the leaderboard, etc.
- Intrinsic motivations
  - Internal to the individual
  - When we behave in some way because we feel a stymulus and a gratification from the very act.
  - In gamification, there are four categories of intrinsic motivations to be considered.



Intrinsic motivations: RAMP

An outcome of the "Self Determination Theory (Deci, Ryan, 2008), the RAMP model (Marczewski) identifies four main types of intrinsic motivators:

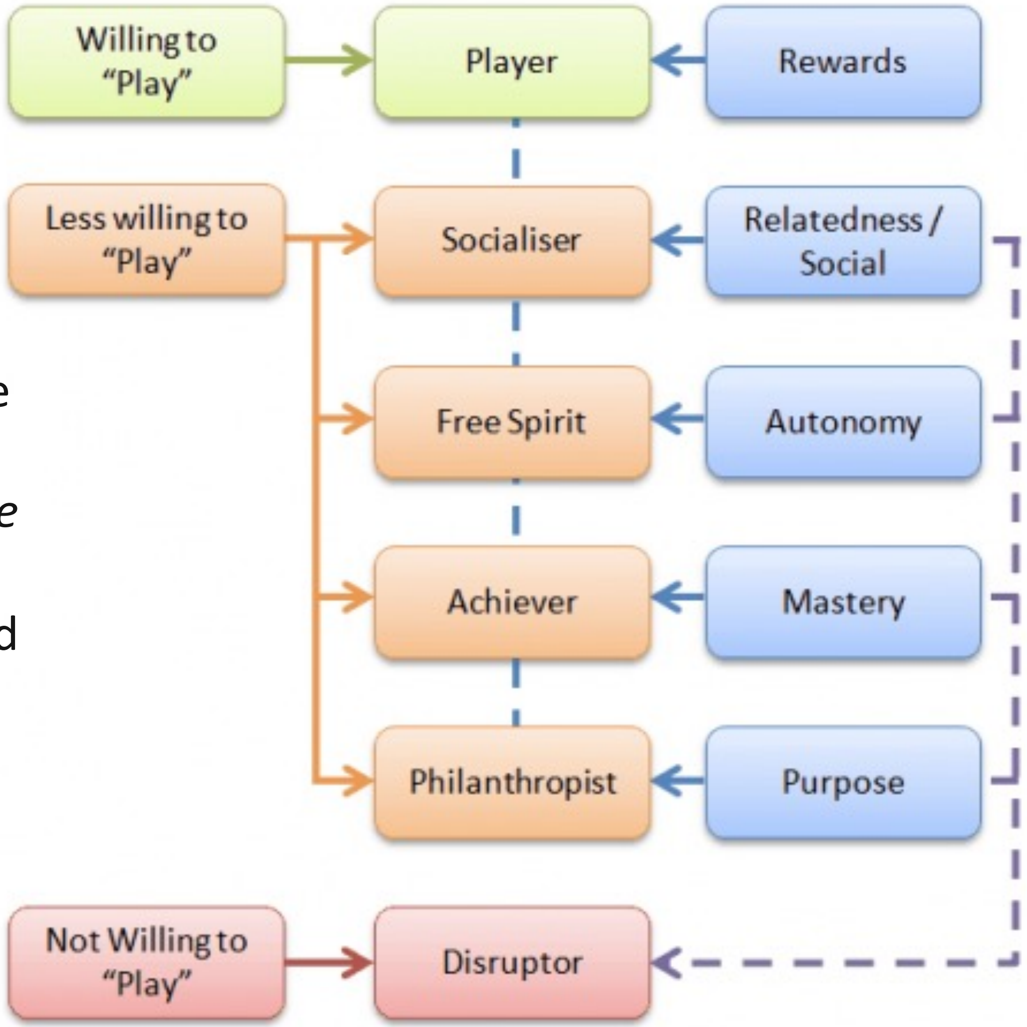
- ***Relatedness***: the desire to be connected to other humans both nearby and remote. Leaderboards, but also chats, forums, etc.
- ***Autonomy***: the feeling of being free and in charge of our decisions. Organizing the activities in quests is better than forcing a precise sequence of rigid actions.
- ***Mastery***: the process of becoming skilled at something. We take pride at being good at something complex or complicated. Pacing in the increase of difficulty of the activities is important as it makes the user feel aware of the progress he/she is making with his/her skills.
- ***Purpose***: the identification of the overall meaning of our activities, especially in relation and interaction with others.



Games and players

Six main types of players, having different motivations:

- ***Players*** are in the game for themselves, to collect *rewards.*
- ***Socializers*** want to interact with others and create connections.
- ***Free Spirits*** are in the game to explore and create.
- ***Achievers*** *want to become good at the mechanics of the game.*
- ***Philantropists*** want to help others and increase the usefulness of the game for the others. Not in the game for themselves
- ***Disruptors*** are in the game for the desire to change it and control it.



References

Jesse Garrett, *The elements of user experience*, New Riders, 2011

Philip Harris, *Data Driven Design*, K&R publications, 2013

A. Cooper, R. Reimann, D. Cronin, *About Face 3*, Wiley, 2009

Alan Cooper, *The inmates are running the asylum*, SAMS, 2004

T. Tullis, B. Albert, *Measuring the User Experience*, Morgan Kaufmann, 2013

Susan Weinschenk, *100 things every designer needs to know about people*, New Riders, 2011

Usability Book of Knowledge, http://www.usabilitybok.org/

David Evans, *Bottlenecks,* Apress (2020).