# User Centered Design of 3D/VR/AR Applications

Dirk Zimmermann, SBS C-LAB
Dirk.Zimmermann@c-lab.ce

## 1. Motivation

3D, Virtual Reality (VR) and Augmented Reality (AR) systems are currently still in an exploratory stage, there are only rare real world applications, let alone systems that penetrate the mass market. Current target users utilize the existing professional systems mainly for specialized tasks like e.g. complex system maintenance or military tasks.
However, there's a great opportunity for these immersive systems to be used by a much larger audience in more consumer oriented settings, not requiring unaffordable hardware or cumbersome interaction techniques.
On the edge from research to real world application, the technology itself moves out of the focus of interest and the actual engineering process, i.e. ensuring the system complies with requirements, quality standards, etc. becomes more important. Manufacturers need a stable and reliable toolset, comprising processes, specification methods and evaluation techniques, to base their development upon.
This position paper outlines basic ideas for these three areas, illustrating how different existing approaches might be adapted for the specific application area of 3D/VR/AR systems.

## 2. Engineering Processes

The process of software engineering stems already from the early times of system design, when everyday users started benefiting form computers and it was not just experts that would actually run a system. It was then that manufacturers realized that projects became larger, were implemented by multiple developers and thus needed a standard procedure for integrating the different phases of software design. The first approaches moved form a mere "code and fix" attitude towards phase models, one of which being the waterfall model. This strictly linear model defined different phases in the process and thus allowed distribution of tasks and also incorporated checkpoints for evaluation after each step in the process. This model is still frequently used in design processes, as it fits well into the process planning world, that also is more or less linear and moves through subsequent steps. However, the actual design process is not exactly reflected by this model. The links between phases (especially the analysis and testing phases) is too limited, i.e. logical links between a task analysis and the testing process at the end are neither assumed not defined. Therefore, the V-model was developed. This model is still a phase model, i.e. different steps are subsequent to each other, yet it also has another dimension that illustrates the process of analysis and specification (abstraction) in contrast to implementation and testing (synthesis).

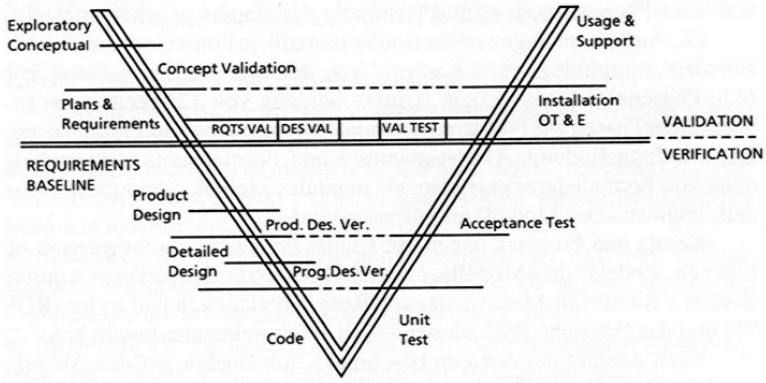


*Fig. 1: V model*

The advantage of this model lies in the connections between the analysis and specification phases (left) and the testing phases (right). A horizontal line indicates that during specification the criteria and goals for this level of analysis are already defined, allowing different levels of testing at the end.
One disadvantage of this model is, that it is still linear and doesn't provide a means to model iterative design processes, e.g. through prototyping. This disadvantage has been overcome by the iterative models, e.g. the spiral model:

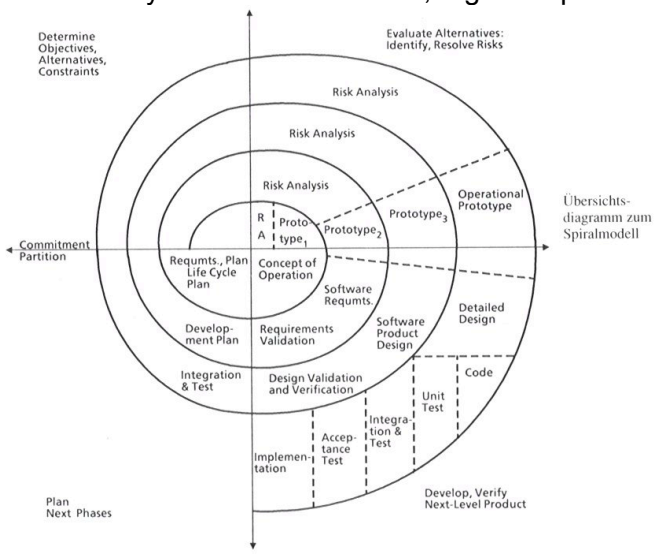


*Fig. 2: spiral model*

This model clearly indicates the cyclic nature of the model, i.e. the same four areas (Plan next Phases, Determine Objectives, Evaluate Alternatives, Develop & Verify) are run through in four iterations, each of them focussing on a different stage in the design process. The two advantages associated with this model are on the one hand the use of prototypes for each stage of the process and on the other the implementation of a risk analysis phase in each stage, i.e. a clear decision point where the feasibility and efficiency of pursuing the project is assessed. However, this model is not as clear as the V model with regard to dependencies between analysis and testing phases.

The most current approach to engineering processes is incremental design, used e.g. in the method of extreme programming. One of the main ideas is a segmentation of the application into manageable chunks and the pursuit of a stepwise process for the smaller

projects. For each project, again, iterations are used to have an incremental optimisation of the solutions until the requirements are met.
Taken together, the main advantages of these three models (v, spiral and incremental) are all beneficial for the design of 3d/VR/AR applications. Due to the novelty of the systems, an iterative approach is needed to continuously optimise, the link between analysis and testing is essential for the non-functional areas of system design, and the segmentation is helpful for making larger scale projects manageable in a distributed team environment.
So for AR applications, an engineering process should provide all of these features in an integrated way.

## 3. Requirements & Specifications

Requirements should be the driving force behind the scope, concept and design of a system. They are gathered during the early phases of engineering and updated and modified throughout the rest of the process.
There are different levels of requirements, some of them being functional, i.e. regarding the features/functions in focus, others being non-functional, gearing towards usage, perception or related criteria of the system.
The relevant requirements for a system should focus on the problems and tasks a user has to solve, not on the actual solution for these. They should define categories and goals the system should meet in the real world. There are already quite a few formats to capture requirements, all of them incorporating different requirement aspects. Some of the relevant aspects are:

- Users: different types of users, their traits, motives, goals and objectives, characteristic criteria describing the expectation towards the actual use of a system
- Tasks: different types of tasks to perform or problems to solve using a planned system, describing the pre/post conditions, flow of events, checkpoints, success/failure criteria etc.
- Context: Different situations or environments in which to use a system, describing the physical (i.e. auditory, visual, thermal, etc.), social (e.g. groups) and cognitive (privacy, time pressure, etc.) environment in which a system will be used.

The respective requirement capture methods are:

- Users: Personas, i.e. a set of fictional, representative user archetypes based on the behaviors, attitudes, and goals of people. Personas have names, personalities, pictures, personal backgrounds, families, and, most importantly, goals; they are not "average" users but specific characters. A persona is a stand-in for a unique group of people who share common goals; at the same time, persona characteristics encompass those of people in widely different demographic groups who may have similar goals.
- Tasks: Use Cases, i.e. a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.
- Context: Usability Context Analysis / Context of Use Analysis

These methods already provide a means to specify a lot of the important characteristics of user, task and context, but they currently stand side by side and are not integrated with each other. For 3D/VR/AR applications, a closer integration and cross reference should be achieved.

## 4. Evaluation

User validation is currently becoming more and more common for professional software and web services. The two most common evaluation frameworks are the ISO 9241 criteria and Nielsen’s Design Heuristics. They both aim at evaluating a given system against

a predefined set of criteria and thus describe their usability. Whereas Nielsen's heuristics are very high level design rules, the ISO 9241 criteria evaluate a system in the context of a specific task or workflow. Both frameworks are suitable for their intended application area, being web sites (heuristics) and office systems (ISO criteria). There is a multitude of evaluation methods and techniques to assess system usability, but as soon as systems are more interactive or the goals are not necessarily task based and hedonistic aspects become more important, these frameworks reach their limits. They have no representation of e.g. fun or joy of use, curiosity, mastery motivation, etc. thus ignoring all the non-functional requirements that are so vital for 3D/VR/AR system design.
Usability Testing, e.g., utilizes tasks and scenarios to lead participants through the different areas of the system and have them evaluate different aspects. For 3D/VR/AR systems, there aren't necessarily tasks to perform, especially when they are geared towards hedonistic requirements as diversion, curiosity, etc.
There are, however, methods and techniques to assess motivational, emotional or cognitive aspects of system use and perception, that have been developed in the area of psychological research, e.g. the repertory grid technique for capturing conceptual/cognitive/emotional assessments without a direct task relation. Their use is still in early stages and the methodology is not yet integrated into the common evaluation framework.
Especially with regard to the different types and levels of user requirements, an evaluation methodology for 3D/VR/AR applications needs to provide techniques to assess all aspects of system use and perception. Technology can assist in doing this, e.g. eyetracking devices can be used to determine the attention to specific UI elements in contrast to the actual use as a proxy for the amount of affordance for this element, or as a non-intrusive means to capture trails of perception and interaction without continuous think-aloud.

## 5. Conclusion

When trying to apply existing processes, methods and criteria of software and usability engineering to the new realm of 3D/VR/AR systems, there are three big points that are not sufficiently covered and require future attention:

- Due to the novelty of the applications and their domains, the engineering process needs to foresee iterations, segmentation and different requirement levels to allow for a close integration of exploration, design and evaluation
- The requirement types used in software design need to address functional and non-functional requirements alike, with a growing emphasis on the non-functional ones, being users' goals, motives, traits, experiences, etc., that need to be known in order to provide users with the appropriate system properties
- Evaluation techniques need to address these non-functional requirements in addition to the task and system centred view that a lot of existing methods are taking. A more systematic and quantitative approach to subjective assessments should be taken in order to provide the required evaluations.

Even though there are single elements already being used in different domains, the goal should be to develop an integrated process, criteria and method framework applicable to the growing area of 3D/VR/AR systems.