


http://first-stage.eu

| | |
|---|---|
| **Project Number** | **688244** |
| **Project Acronym** | **first.stage** |
| **Project Title** | **Fast and Easy Previsualisation for Creative Industries** |
| Deliverable Number | D1.5 |
| Deliverable Name | Core Functionality |
| Type | Report |
| Status & Version | Public, Version 2.0 |
| Work Package Contributing to the Deliverable | WP1 |
| Work Package / Task Responsible | Next Limit SL / University of Bremen |
| Other Contributors | InfoConsult Gesellschaft für Informationstechnik mbH |
| Authors | Thomas Fröhlich, Thomas Muender, Dr. Dirk Wenig |
| Keywords | Previs, previsualisation, creative industries, functionalities, toolset |

The research leading to these results has received funding from the European Union's H2020 Programme for research, technological development and demonstration under grant agreement n° 688244

# *Executive Summary*

***Aim of the Deliverable.*** *This deliverable was set up to define the core functionalities of the first.stage software toolset.*

***Brief Description of the Sections of the Document.*** *After an introduction to the deliverable in section 1, the requirements collected in D1.1 to D1.4 are reviewed, sorted, prioritized, and presented in section 2. In section 3, the core functionalities are presented along with a list of the demonstrators and matched to the individual demonstrators.*

***Mayor Achievements.*** *Presentation of core functionalities and application specific functionalities based on the collection of requirements from D1.1 to D1.4 and an internal workshop. Milestone MS2 ("Core functionalities defined") is accomplished with this document.*

***Summary of the Conclusions Obtained.*** *20 core functionalities and 4 additional functionalities are defined referencing back to the requirements.*

  

# Content



  

# 1 Introduction

The goal of the first.stage project is to research, design, develop and evaluate tools that improve previsualisation in film, animation, and the performing arts. In first.stage, these tools are demonstrators that contain of different hardware and software, representing the use of next-gen previs tools, implementing natural interaction patterns to make this technology accessible for a large group of users.

In order to create tools for users coming from a variety of technical and artistic backgrounds, we must understand who our users are, how they currently work and what they envision for previs tools of the next generation. Thus, the elicitation of requirements is very important considering the variety of tasks and users in these diverse production types.

The goal for this document is to define core functionalities for the first.stage demonstrators, which are "those functionalities of the product without which, the product is not useful for the users" [1]. Core functionalities are thus to distinguish from ancillary functionalities, which supplement core functionalities. The latter, if not implemented, do not hinder productivity, but may have an impact on the convenience of use.

For the collection of requirements, we followed a user-centric approach, consisting of interviews with practitioners, created use cases, and scenarios. We applied each method to each of the application domains, and extracted requirements from all domains, created an overall workflow for previs, categorized the requirements into application specific and common categories, and further prioritized them in a peer-review process with all partners in order to come to a final set of core functionalities for practitioners. In a workshop, we discussed, elucidated, and prioritized further requirements within the consortium to make sure that no crucial functionality is missed due to the novelty of our set of tools from the point of view of end users.

The outcome of this process is a set of core functions, that we treat a "must-have" components in the development of future demonstrators so that we can assure a high usability for the main tasks for all application domains throughout all demonstrators.

# 2 Requirements

In order to identify the core functionalities for a previs software, requirements have to be collected from users and other stakeholders to generate a common picture that integrates the views of the different application areas. However, aggregating the diverse requirements into common picture for the identification of core functionalities is hard because there is a wide variety of preferences and use cases that have to be considered. In order to solve this problem, a structured approach is needed that filters and sorts the diverse data to make sure that the core of the system supports all crucial tasks and expands around these core functions to support adjacent functions to satisfy the needs of all application areas.

Our approach consists of the following steps: a) extraction of user requirements from interviews, personas, use cases, scenarios, and functional requirements, b) additional expert-based requirements from the first.stage consortium, c) workflow generation, d) prioritisation, e) categorisation, and f)

  

extraction of hardware requirements. This forms the foundation for core functionalities, which will be introduced in the following sections.

In the following sections, we present user requirements from all application areas in a condensed form, present the results of a workshop where complementary requirements were generated by experts to complement the user requirements, and a circular workflow which was created on the basis of all collected requirements. Further, the results of the prioritisation of these user requirements, a categorisation into common, overlapping, and application specific items, and a supplementary list of hardware requirements are presented.

## 2.1 Review of Requirements

In this section, we review the content of the Deliverables D1.1 to D1.4: Requirements for animation production, TV/commercial/film production, stage production, and visual effects to give an overview of the overall collected information. Each of the application partners Your Mom GmbH (YoMo, Arx Anima), OO Theater und Orchester GmbH (TOG), Vogel Audiovision Gesellschaft mbH (Vogel), and Next Limit SL (NL) created one of these documents where they conducted interviews with possible users of a previs software to find who the users are (e.g. directors, producers, and other staff), what their preferences and current workflows look like, and what they require in a previs software.

### 2.1.1 Summary of Requirements for Animation (D1.1)

In the Deliverable D1.1 the requirements for the application area “animation” were developed. The requirements are based on interviews with a professional animation supervisor, layout supervisor and editing supervisor.

In order to account for the project-oriented work in the animation area one of the key requirements is the ability to create and work on projects within the software. Such projects should be defined by general information and contain all information on scenes, shots, layout, animations, cameras and effects. As such projects are team based it is also required to share the project with several persons and to work simultaneously.

Starting a new project, it is required to import any kind of media to represent the ideas of creative staff. This includes videos, images text and websites but also 3D models and assets from previous projects. It has to be possible to arrange these items spatially and temporally and define relationships. A state of the art timeline with simplified interface is required for this.

As an animation project consists of many parts e.g. Scenes and Shots it is necessary to organize the project using these building blocks. Scenes should represent the 3D arrangement of objects and shots should represent a view on the scene defined by a specific camera. Scenes and shots should also be arranged on the timeline.

For communication between team members and to express ideas it is required to attach notes to all objects and share these with other people. Notes making should be possible as text and voice recording.

In order to integrate the software into the workflow it is a key requirement to offer import and export functionality to other software used in the industry.

  

Production of an animation project after the previs stage takes time, employees and money. Being able to plan these costs can be a major benefit. Therefore, it is necessary to export lists of assets (3D models, animations etc.) from the previs software.

To create detailed and persuasive animations motion capture is state of the art. The integration of this technique in an easy to use manner is required.

Furthermore, the deliverable describes advantages of the use of virtual reality for sketching ideas in 3D and lay-outing of scenes.

### 2.1.2 Summary of Requirements for Film/Commercial Productions (D1.2)

In the Deliverable D1.2 the requirements for the application area “film production” were developed. The requirements are based on interviews with a professional producer, set designer cinematograph, and two directors.

Starting the creative part of a project in the film industry is mostly done by drawings and scribbles e.g. Storyboard. Sketching ideas should also be possible as a starting point in a previs software. In order to build a set based on the ideas CAD software is used to plan and visualize the sets. The import of CAD data into the previs software to layout a scene is important. In layout it is also necessary to have access to a library of assets to furnish the scene.

Planning of actors is important for film production. Therefore, it is required to be able to select different character models, place them in the scene and visualize their motions. This could be done by animation, simulation along paths. An important part of a film project is the camera. Different internal settings like lenses, zoom and depth of field have to be configurable. Also, the positions and motions of the camera for the shots should be able to be planned and visualized. Completing the scene setup, it is required to modify lights. Adding and positioning lights and choosing colour as well as general light moods e.g. sunset, dawn, indoor, outdoor is necessary.

Modern films heavily rely on computer generated images (CGI). Planning how a film will look after adding these elements is very important. Therefore, it is required to add CGI elements to a scene and position them in the image view. Same applies for the integration of green screen video elements.

Stunt performances can be very expensive because of planning, testing/ training and safety management. Previs could help with the planning and safety management by simulating the stunts beforehand. For this to be feasible it is required to have accurate physics simulation.

In order to iterate fast and be able to evaluate the look of the film it is required to have real time film export of the project in sufficient rendering quality. To show the results for internal advertising and to acquire funding there should also be the option to produce photorealism rendering with longer rendering times. For managing purposes, it is also required to export item lists and timing info of the scenes.

In order to work in the office as well as on site and to match the habits of creative personal it is required for the software to work on a range of different hardware from tablets to desktop work stations. The main hardware requirement is to work on a standard laptop. To make creative use the software and adapt to it fast it is necessary that the software is easy to use and comes with a clean and understandable user interface.

  

### 2.1.3 Summary of Requirements for Theatre/Stage Productions (D1.3)

In the Deliverable D1.3 the requirements for the application area “theatre/stage production” were developed. The requirements are based on interviews with a theatre managing director, intendant, theatre CTO, artistic director, set /stage and costume designers, head stage engineer and the head of light department.

For theatre productions the key element is the stage which in contrast to film production is a fixed asset. Importing and working with an accurate 3D model of the real stage is required. Import of the stage model as well as imports in general should be based on standard file formats of the industry. This also applies for the export functionality of the software as well as the communication with other software via standard protocols.

It should also be possible to represent the technical infrastructure and account for the physical limitations of the machinery. This is necessary to illustrate the opportunities and (physical) restrictions of the imported stage. Lay-outing a set or scene highly depends on the stage so it is required to be done directly on the imported stage model with regard to its limitations. In order to validate a set design in terms of visibility from all seats in the audience and that no one can look behind the set it is required to test lines of sight from different positions. To build a set, material costs and weight have to be taken into account. The software should be able to export lists of assets with size and material information to calculate these factors.

Another part that is a fixed asset for stage production is the light setup. The position, turning angles, and colour options should be representable in a previs software. This would enable for easy light planning with contributes to the overall perception of a stage setup. In addition, sometimes light projections are used as part of the stage design. It is required to plan with such elements in previs. Using elements from previs for digital projections would also need optional photorealistic rendering.

As digital artefacts get used more in modern stage productions it is also necessary to integrate green screen video elements as well as motion capture into the previsualisation of such productions.

Many people are involved in the development of a stage production. To account for their distributed work and to support communication the software requires the ability to work on the same project with several people at the same time. Being able to share the set models with suppliers should simplify the communication of creative ideas. To provide this, the software should have a presentation mode with an automatically updated project version.

Managing multiple stage productions requires to plan the stage use in terms of storage and suspension for multiple sets. This should be represented by the ability to load multiple scenes on the stage model with regard to their need for space on stage and the suspension area above the stage. When two scenes are loaded, conflicting parts should be highlighted.

To be used by creatives who often have not much technical knowledge, the software requires an easy to use interface. The deliverable proposes specialized modules for different tasks and different layers of complexity. Furthermore, the software should run on standard mobile and desktop hardware.

  

### 2.1.4 Summary of Requirements for Visual Effects (D1.4)

In the Deliverable D1.4 the requirements for the application area “visual effects” (VFX) were developed. The requirements are based on interviews with a (short)film producer, two creative directors of game studios, and a research and development director.

One of the main requirements for the previsualisation of visual effects is the ability to integrate the software into the existing workflow. This can be achieved by import and export functionality to industry standard file types for 3D models, texture, light, etc. In addition, the assets created during previous e.g. animations should be reusable in later production stages.

In order to create visual effects keyframe animation as well as physics simulation for cloth, soft bodies and fracture are required. The interface and parameter for such simulations should be easy to use and to adapt to create the desired effects.

Evaluation of the visual aspect and communicating ideas are key to previs. Therefore, the software needs an immersive presentation mode.

In order to be used in the industry a previs software also requires a comprehensive documentation and it needs to be sold under standard licensing models.

## 2.2 Requirements Workshop

Having extracted user requirements from interviews, we additionally collected functional requirements in a workshop with the whole first.stage consortium that complement the user requirements. We used the outcome of this workshop to gain inside into common requirements and make a first prioritisation. With this approach we extend the view from the user perspective to include experience and preferences of domain experts from the application partners as well as discussed feasibility and technical aspects. We used the MoSCoW [2] method for the prioritisation into must have, should have, could have, and won’t have requirements. The results are presented in Table 1.

*Table 1: MoSCoW analysis of the requirements from the requirements workshop with all first.stage partners*

| | **Must** | **Should** | **Could** | **Won’t** |
|---|---|---|---|---|
| **System architecture** | - Unity as a base platform<br>- MovieStorm (MoSt) software on top<br>- Use of plugins provided by third-parties and partners, exposed in the stand-alone tool<br>- Cloud-based assets database | | | |
| **Compatibility with other tools** | - Import/export formats: FBX, Alembic, etc. | - Other formats: USD, etc. | | |
| **Type of files** | - Scene: OBJ, STL, FBX, Alembic, etc.<br>- Textures/Images: JPG, PNG, EXR, etc. | | | |

  

| | | | | |
|---|---|---|---|---|
| | - "first.stage scene format" FSS (XML) | | | |
| **Import /Export** | - Import: Meshes, Materials, Rigs, Cameras<br>- Export: Scene information, sequences, camera information, caches | | | |
| **Management of assets** | - Assets database | - Cloud-based repository | | |
| **Creation/Editing** | - Basic rigid object animation<br>- Live mocap recording | - Sketch tools | | |
| **Workflow and User Experience** | - Simplicity<br>- Based on established workflows (seamless integration)<br>- Editing (shots, sequences, …)<br>- Iterative / versioning functions | | | |
| **User cooperation** | - Role-based interface/editing (different levels of control and access to functions)<br>- Tracking / history of changes (notifications) | - In-game communication | | |
| **Graphic user interface** | - New, simple interface based on existing interface paradigms | | | - Complex tool full of buttons |
| **Natural user interface** | - Object manipulation by gestures<br>- Camera control<br>- Physically-based control<br>- Real improvement over mouse and keyboard | - Voice control | | |
| **Layout** | - Layout is KEY for first.stage<br>- Sketching import, 3D model layout, POV definition, measures | - Physical interface<br>- Scribbling as a placeholder (mock-up) | - Scanning of real props | |
| **Sound** | - Import sound files<br>- Timeline editing<br>- Multiple tracks | - Event-driven<br>- Triggers | - 3D sound attached to objects | |
| **Posing** | - Static pose recording<br>- Libraries (versions of the same asset)<br>- Mesh versions of skinned meshes with bones | - Posing functions | - Explore Unity's options (mesh, bones, rigging) | |
| **Rigging** | - Import rigged characters | | | |

  

| | | | | |
|---|---|---|---|---|
| | - Grouping / parenting functions | | | |
| **Animation** | - Import animated / baked geometry<br>- Functions to create, edit, re-time animations<br>- Keyframe animation of static objects<br>- Apply library poses and animations to rigged objects | | | - Facial animation<br>- Keyframing<br>- Bones animation |
| **Simulation** | - Real-time engine<br>- Looped animations<br>- Aimed animations | - As fast and interactive as possible<br>- Hi-hi physics simulation<br>- Editable and art directable | | |
| **Lighting** | - Open library including realistic light sources<br>- Compatibility with existing light libraries<br>- Editing basic parameters of the light source<br>- Animate light sources (intensity variation) | - Animate flaps | - Compatibility with light consoles, light linking | |
| **Camera** | - Aperture and focus length<br>- Keyframe camera animation / movement<br>- Camera rig (stereo rig, e.g.) for export purposes | - Standard existing camera setups, cranes, steadicam, physical information, etc.<br>- Gesture control for cameras | - Post process effects | - Optics simulation |
| **Rendering** | - Shaders in Unity: sufficient realistic light models and rendering for real-time interaction | - Material description is the same for real-time shaders and Octane as an offline renderer | - High quality rendering for realistic representation of light (offline process) | |
| **Editing** | - Ability to overview shot sequences in the timeline (mixed media overview) as a “folder system” of projects, sound editing, trimming (linear editing) | - Non-linear editing<br>- Muting shots<br>- CG / real tagging<br>- Notes making<br>- Fade-to-color effects | | - Comprehensive editing toolset |

  


*Figure 1: Collection of requirements ordered by task. Origin of requirement is colour coded: white – general, green – theatre / stage production, red – animation, orange – film / commercial, blue – special effects*

## 2.3 Initial set of user requirements and workflow

On the basis of the deliverables D1.1 to D1.4 and the requirements workshop we extracted user requirements by coding the interviews, scenarios, and use cases in each document. Must and Should parts of the MoSCoW analysis were added to the set of requirements. In addition, we added requirements that were mentioned in the interviews but did not appear in the requirements reports. Then, we created an index card for each of the items. We used color-coding for identification of the source for this requirement:

Animation | Film Production | Theatre / Stage | Special Effects | Workshop / Interviews

From the whole set of requirements, we sorted all cards in individual workflow steps: Import, setup/sketch, layout, shots, camera, light, sound, posing, animation, motion capture, simulation, rendering, and export. The workflow steps were then organized in circular order to resemble an overall workflow of the system functionality. This workflow is presented in Figure 1. The items in the middle of the circle are general requirements, that cannot directly be assigned to one of the identified categories.

## 2.4 Prioritisation of User Requirements

Having collected, sorted and organized all the 118 user requirements, we started a prioritisation process in order to identify the most relevant items for the implementation in the demonstrators. For the prioritisation, we decided to let each the first.stage application area partners rank all the requirements. The advantage with this is, that these partners can rate each requirement in the context of their specific expertise and preferences regarding previs in general and for the desired outcome of the project.

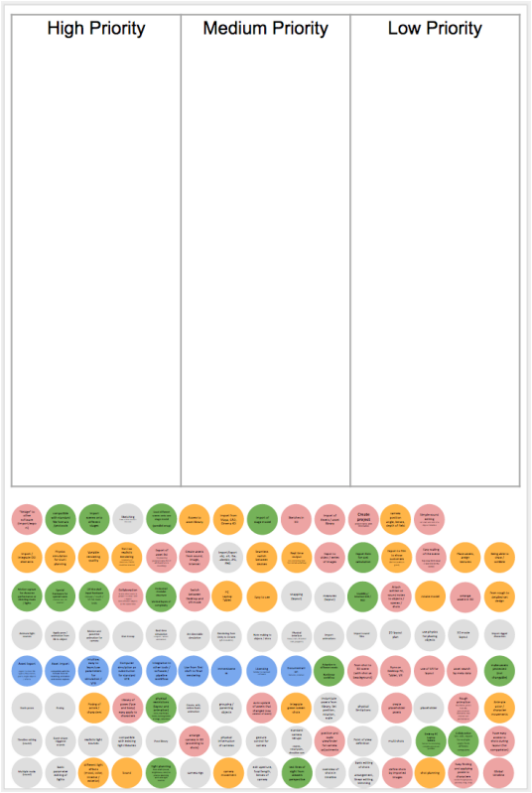


*Figure 2: Unsorted prioritisation document*

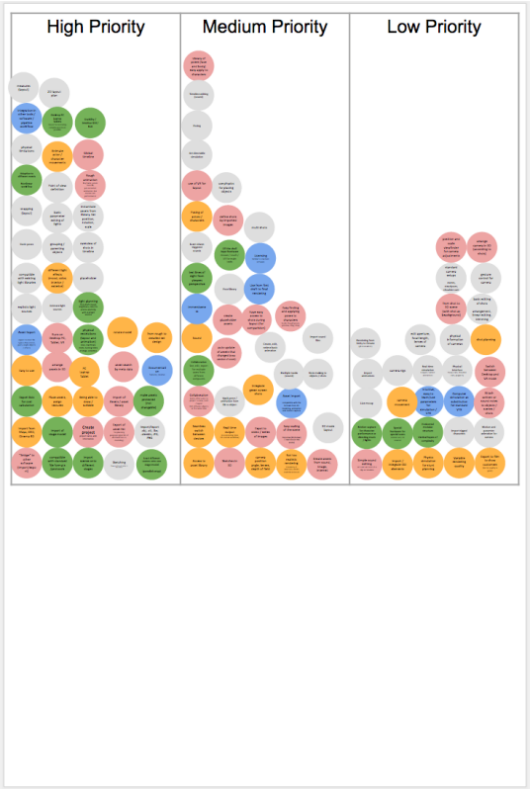


*Figure 3: Completed prioritisation document*

The process was organized as follows: First, all requirements were transferred into a text document, including a table with 3 columns: high priority, medium priority, and low priority, see Figure 2 and Figure 3. A copy of this document was distributed to each partner for ranking. After each partner completed the ranking process, each requirement was assigned a score: For a high priority item three points, medium priority two points, and low priority one point.

The results of all four completed prioritisations were then transferred into a table, including the overall score and an average score, see Table 2, where origin indicates from which application area the individual requirement was collected, animation, film, stage, and VFX contain the score from each of the partners, sum represents the total score, and average the calculated average.

  

*Table 2: Requirement prioritisation by the partners*

| | Requirement description | Origin | Animation | Film | Stage | VFX | Sum | Average |
|---|---|---|---|---|---|---|---|---|
| **1** | "Bridge" to other software (import/export) | A | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **2** | Arrange assets in 3D | A | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **3** | Compatible with standard file formats / protocols | S | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **4** | Create project (project name, add. Information) | A | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **5** | Import of assets / asset library | A | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **6** | Import/Export .obj, .stl, .fbx, .alembic, .JPG, .png | G | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **7** | Instantiate assets from library. Set position, rotation, scale | G | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **8** | Integration in other tools / software / pipeline workflow | V | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **9** | Place assets, assign textures | F | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **10** | Rotate model | F | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **11** | Rough animation (Keyframe, poses from lib, pre-recorded animation, live motion suit performance) | A | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **12** | Static poses | G | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **13** | Usability / Intuitive GUI / NUI | S | 3 | 3 | 3 | 3 | 12 | 3,00 |
| **14** | 3D mode layout | G | 3 | 3 | 2 | 3 | 11 | 2,75 |
| **15** | Access to asset library | F | 3 | 3 | 2 | 3 | 11 | 2,75 |
| **16** | Apply pose / animation from lib to object | G | 3 | 3 | 2 | 3 | 11 | 2,75 |
| **17** | Asset Import, compatible with file formats from 3D modeling, animation and motion capture | V | 3 | 3 | 2 | 3 | 11 | 2,75 |
| **18** | Create, edit, retime basic animation | G | 3 | 3 | 2 | 3 | 11 | 2,75 |
| **19** | Basic parameter editing of lights | G | 3 | 2 | 3 | 3 | 11 | 2,75 |
| **20** | Different light effects (mood, colour, interior / exterior) | F | 2 | 3 | 3 | 3 | 11 | 2,75 |
| **21** | Easy to use | F | 2 | 3 | 3 | 3 | 11 | 2,75 |
| **22** | Import from Maya, CAD, Cinema 4D | F | 3 | 3 | 3 | 2 | 11 | 2,75 |
| **23** | Point of view definition | G | 3 | 3 | 3 | 2 | 11 | 2,75 |
| **24** | Animate actor / character movements | A | 3 | 3 | 3 | 2 | 11 | 2,75 |
| **25** | Arrange camera in 3D (according to shots) | A | 3 | 3 | 1 | 3 | 10 | 2,50 |
| **26** | Camera movement | F | 3 | 3 | 1 | 3 | 10 | 2,50 |
| **27** | Auto update of assets that changed (new version of asset) | A | 3 | 3 | 2 | 2 | 10 | 2,50 |
| **28** | Camera position, angle, lenses, depth of field | F | 3 | 3 | 2 | 2 | 10 | 2,50 |
| **29** | Easy scaling of the scene (top view (bird view) -> real size (in the scene)) | A | 3 | 2 | 2 | 3 | 10 | 2,50 |
| **30** | Export to video / series of images | A | 2 | 3 | 2 | 3 | 10 | 2,50 |
| **31** | Library of poses (face and body) easy apply to characters | A | 2 | 3 | 2 | 3 | 10 | 2,50 |
| **32** | Pose library | G | 3 | 3 | 2 | 2 | 10 | 2,50 |
| **33** | Posing | G | 3 | 3 | 2 | 2 | 10 | 2,50 |
| **34** | Posing of actors / characters | F | 3 | 3 | 2 | 2 | 10 | 2,50 |
| **35** | Real time output (no rendering time for normal workflow) | F | 2 | 3 | 2 | 3 | 10 | 2,50 |
| **36** | Animate lights sources | G | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **37** | Asset Export (export to most file types, especially .fbx, scenes, parts, single objects / effects) | V | 2 | 2 | 3 | 3 | 10 | 2,50 |
| **38** | From rough to detailed set design | F | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **39** | Grouping / parenting objects | G | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **40** | Overview of shots in timeline | G | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **41** | Import scenes onto different stages | S | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **42** | Load different scenes onto one stage model (parallel setup) | S | 3 | 2 | 3 | 2 | 10 | 2,50 |
| **43** | Camera rigs | G | 3 | 3 | 1 | 2 | 9 | 2,25 |
| **44** | Edit aperture, focal length, lenses of camera | G | 3 | 3 | 1 | 2 | 9 | 2,25 |

| | Requirement description | Origin | Animation | Film | Stage | VFX | Sum | Average |
|---|---|---|---|---|---|---|---|---|
| **45** | Export to film to show customers (better quality is good) | F | 3 | 3 | 1 | 2 | 9 | 2,25 |
| **46** | Intuitive, easy to learn/use parameters for simulation/ VFX | V | 2 | 3 | 1 | 3 | 9 | 2,25 |
| **47** | Motion and parameter animation for camera | G | 3 | 3 | 1 | 2 | 9 | 2,25 |
| **48** | Collaboration (invite other users to project to view / edit / export; Edit different objects at the same time) | A | 3 | 2 | 2 | 2 | 9 | 2,25 |
| **49** | Collaboration (See, edit, export for multiple users from different computers) | S | 3 | 2 | 2 | 2 | 9 | 2,25 |
| **50** | Create assets from sound, image, internet | A | 3 | 3 | 2 | 1 | 9 | 2,25 |
| **51** | Easy finding and applying poses to characters (scrub through poses, preview, drag, drop) | A | 2 | 3 | 2 | 2 | 9 | 2,25 |
| **52** | Not too realistic rendering (Creativity of directors / DOPs could be reduced) | F | 3 | 1 | 2 | 3 | 9 | 2,25 |
| **53** | Note making to objects / shots | G | 3 | 1 | 2 | 3 | 9 | 2,25 |
| **54** | Use physics for placing objects | G | 2 | 2 | 2 | 3 | 9 | 2,25 |
| **55** | 2D layout plan | G | 3 | 1 | 3 | 2 | 9 | 2,25 |
| **56** | Adaptive to different needs, nonlinear workflow | S | 3 | 2 | 3 | 1 | 9 | 2,25 |
| **57** | Asset search by meta data | A | 2 | 3 | 3 | 1 | 9 | 2,25 |
| **58** | Being able to draw / scribble | F | 3 | 2 | 3 | 1 | 9 | 2,25 |
| **59** | Export of asset list (for planning purposes give list of all characters to modeling) | A | 2 | 2 | 3 | 2 | 9 | 2,25 |
| **60** | Global timeline | A | 2 | 2 | 3 | 2 | 9 | 2,25 |
| **61** | Import of stage model | S | 1 | 3 | 3 | 2 | 9 | 2,25 |
| **62** | Measures (layout) | G | 1 | 2 | 3 | 3 | 9 | 2,25 |
| **63** | PC, Laptop, Tablet | F | 1 | 3 | 3 | 2 | 9 | 2,25 |
| **64** | Placeholder | G | 3 | 2 | 3 | 1 | 9 | 2,25 |
| **65** | Realistic light sources | G | 1 | 3 | 3 | 2 | 9 | 2,25 |
| **66** | Basic editing of shots / arrangement, linear editing, trimming | G | 3 | 2 | 1 | 2 | 8 | 2,00 |
| **67** | From shot to 3D scene (with shot as background) | A | 3 | 2 | 1 | 2 | 8 | 2,00 |
| **68** | Import / integrate CGI elements | F | 3 | 3 | 1 | 1 | 8 | 2,00 |
| **69** | Import animations | G | 3 | 2 | 1 | 2 | 8 | 2,00 |
| **70** | Import rigged characters | G | 3 | 3 | 1 | 1 | 8 | 2,00 |
| **71** | Live mocap | G | 3 | 2 | 1 | 2 | 8 | 2,00 |
| **72** | Position and scale viewfinder for camera adjustments | A | 2 | 3 | 1 | 2 | 8 | 2,00 |
| **73** | Shot planning | F | 3 | 2 | 1 | 2 | 8 | 2,00 |
| **74** | Create placeholder assets | A | 3 | 2 | 2 | 1 | 8 | 2,00 |
| **75** | Have easy access to shots during layout (for comparison) | A | 2 | 2 | 2 | 2 | 8 | 2,00 |
| **76** | Immersiveness | V | 3 | 2 | 2 | 1 | 8 | 2,00 |
| **77** | Multi shots | G | 3 | 2 | 2 | 1 | 8 | 2,00 |
| **78** | Sound | F | 3 | 1 | 2 | 2 | 8 | 2,00 |
| **79** | Compatible with existing light libraries | G | 1 | 2 | 3 | 2 | 8 | 2,00 |
| **80** | Computer simulation as substitution for standard VFX | V | 2 | 2 | 1 | 3 | 8 | 2,00 |
| **81** | Documentation, features, creative | V | 1 | 3 | 3 | 1 | 8 | 2,00 |
| **82** | Horizontal modular structure, vertical layers of complexity | S | 2 | 2 | 1 | 3 | 8 | 2,00 |
| **83** | Light planning from draft (mood, brightness, colour) to precise planning with real light sources | S | 1 | 3 | 3 | 1 | 8 | 2,00 |
| **84** | Physical restrictions (layout and animation), sizes, machinery, limits, turning limits of stage, velocities | S | 1 | 2 | 3 | 2 | 8 | 2,00 |
| **85** | Runs on Desktop PC, Tablet, VR | A | 1 | 3 | 3 | 1 | 8 | 2,00 |
| **86** | Sketching (Draw, scribble directly in the scene) | G | 3 | 1 | 3 | 1 | 8 | 2,00 |
| **87** | Snapping (layout) | G | 1 | 2 | 3 | 2 | 8 | 2,00 |

| | Requirement description | Origin | Animation | Film | Stage | VFX | Sum | Average |
|---|---|---|---|---|---|---|---|---|
| **88** | Use of VR for layout | A | 3 | 2 | 2 | 1 | 8 | 2,00 |
| **89** | Desktop PC, Laptop, Tablets, Server for rendering complex processes possible | S | 1 | 2 | 3 | 1 | 7 | 1,75 |
| **90** | Motion capture for character performance or directing music / lights | S | 2 | 2 | 1 | 2 | 7 | 1,75 |
| **91** | Real-time simulation (looped / aimed animations) | G | 2 | 2 | 1 | 2 | 7 | 1,75 |
| **92** | Variable rendering quality | F | 1 | 3 | 1 | 2 | 7 | 1,75 |
| **93** | Art-directable simulation | G | 1 | 2 | 2 | 2 | 7 | 1,75 |
| **94** | Import sound files | G | 2 | 1 | 2 | 2 | 7 | 1,75 |
| **95** | Sketches in 3D | A | 3 | 1 | 2 | 1 | 7 | 1,75 |
| **96** | Use from first draft to final rendering | V | 1 | 2 | 2 | 2 | 7 | 1,75 |
| **97** | Export lists for cost calculation | S | 1 | 1 | 3 | 2 | 7 | 1,75 |
| **98** | Physical limitations | G | 1 | 2 | 3 | 1 | 7 | 1,75 |
| **99** | Attach written or sound notes to objects / scenes / shots | A | 2 | 1 | 1 | 2 | 6 | 1,50 |
| **100** | Physical information of cameras | G | 1 | 2 | 1 | 2 | 6 | 1,50 |
| **101** | Physics simulation for stunt planning | F | 1 | 2 | 1 | 2 | 6 | 1,50 |
| **102** | Simple sound editing (set start and end of a clip on the timeline) | A | 2 | 1 | 1 | 2 | 6 | 1,50 |
| **103** | Standard camera setups, cranes, steadicam, shoulder cam | G | 1 | 3 | 1 | 1 | 6 | 1,50 |
| **104** | Define shots by imported images | A | 2 | 1 | 2 | 1 | 6 | 1,50 |
| **105** | Event driven triggered sounds | G | 2 | 1 | 2 | 1 | 6 | 1,50 |
| **106** | Integrate green screen shots | F | 1 | 2 | 2 | 1 | 6 | 1,50 |
| **107** | Licensing, similar to standard software | V | 1 | 2 | 2 | 1 | 6 | 1,50 |
| **108** | Multiple tracks (sound) | G | 2 | 1 | 2 | 1 | 6 | 1,50 |
| **109** | Off the shelf input hardware, mouse, touch, VR for main tasks | S | 1 | 2 | 2 | 1 | 6 | 1,50 |
| **110** | Seamless switch between devices | F | 1 | 2 | 2 | 1 | 6 | 1,50 |
| **111** | Test lines of sight from viewers perspective | S | 1 | 2 | 2 | 1 | 6 | 1,50 |
| **112** | Timeline editing (sound) | G | 2 | 1 | 2 | 1 | 6 | 1,50 |
| **113** | Make assets protected (not changeable) | S | 1 | 1 | 3 | 1 | 6 | 1,50 |
| **114** | Special hardware for special cases, motion suit, 3D scanner | S | 2 | 1 | 1 | 2 | 6 | 1,50 |
| **115** | Gesture control for camera | G | 2 | 1 | 1 | 1 | 5 | 1,25 |
| **116** | Physical interface, heavy cam, shoulder cam, puppetry | G | 1 | 2 | 1 | 1 | 5 | 1,25 |
| **117** | Switch between Desktop and VR mode | A | 2 | 1 | 1 | 1 | 5 | 1,25 |
| **118** | Rendering from Unity to Octane (photorealistic) | G | 1 | 1 | 1 | 1 | 4 | 1,00 |

## 2.5 Requirements Categorisation

On the basis of the prioritized requirements, the next step was to identify completely overlapping items, and those only shared by two or three application areas. Further, some of the requirements are overlapping in meaning and those have to be found and merged into one requirement. Figure 4 illustrates common requirements (dark blue) which are shared by all application areas, overlapping requirements (light blue) shared by two to three areas, and application specific requirements (white).

  

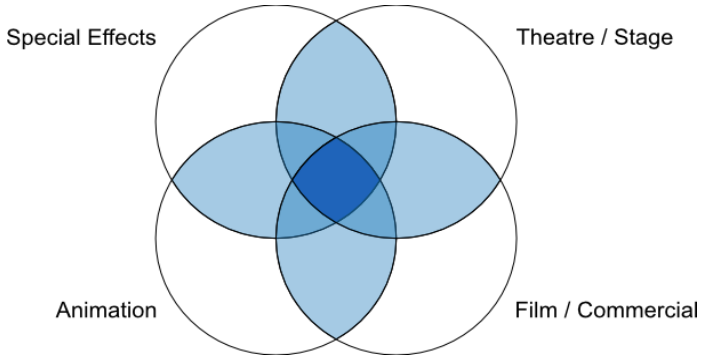


*Figure 4: Illustration of application areas and possible overlaps for feature requirements*

The following sections present the results of the categorisation and accumulation of similar requirements. Each requirement was assigned with an ID, an expressive name, source, and extended description.

### 2.5.1 Common Requirements (CR)

The following requirements have initially been provided by all the partner either in the deliverables D1.1 to D1.4 or in the workshop, or they have been prioritised with highest priority by all the partners – except the requirement for compatibility with industry standard protocols which has been highly prioritised but is not in the scope of the project (AR26, see Section 3.3). Requirements without priority have been added after the prioritisation process and weighted in the derivation of the core functionalities.

*Table 3: first.stage Common Requirements*

| ID | Name | Priority | Source | Description |
|---|---|---|---|---|
| **CR1** | Import from industry standard file types | 3 | D1.1 – D1.4, Workshop | Functionality to import data from file types like: .obj, .stl, .fbx, .alembic, .jpg, .png, .mp3, etc. |
| **CR2** | Import from other applications (Maya, CAD, Cinema 4D) | 3 | D1.1 – D1.4, Workshop | For workflow integration import from other industry standard software is required. |
| **CR3** | Import of Assets | 3 | D1.1 – D1.4, Workshop | Import functionality for assets like: 3D models, textures |
| **CR4** | Import of rigged characters | 2 | D1.1, Workshop | Import functionality for characters that are rigged. Necessary to apply animations to characters. |
| **CR5** | Import of sound files | 1,75 | D1.1, Workshop | Import of sound files like .mp3, .wav |
| **CR6** | Import of animations | 3 | D1.1, Workshop | Import of predefined animations |
| **CR7** | Sketching | 2 | D1.1, D1.2, Workshop | Ability to draw as a starting point. Or import of drawings, storyboard. |
| **CR8** | Instantiate asset; Set position, rotation and scale | 3 | D1.1 – D1.4, Workshop | Create one or more copies of imported assets and position it in a scene. |
| **CR9** | Placeholder assets | 2 / 2,25 | Workshop | Position asset in scene that can be replaced by another later on. Visually identifiable as placeholder. |
| **CR10** | Layout in 3D | 2,75 | D1.1 – D1.4, Workshop | Arrange objects (assets) in 3D view. |
| **CR11** | 2D layout plan | 2,25 | Workshop | Arrange objects (assets) in a 2D top down view. Layout plan. |
| **CR12** | Grouping, Parenting | 2,5 | Workshop | Group multiple objects together. Changes to the group apply to all objects of the group. Make object |

  

| | | | | |
|---|---|---|---|---|
| | | | | child of another object. Changes to parent affect child. |
| **CR13** | Snapping | 2 | Workshop | Objects snap to a position in relation to other objects that are already in the scene. |
| **CR14** | Measures | 2,25 | Workshop | Display coordinates of objects and distances between objects. |
| **CR15** | Physical layout restrictions | 2 | Workshop, D1.3 | Using physics for placing objects. No intersection of objects. Place like they would have fallen. |
| **CR16** | Integration of green screen elements | 1,5 | D1.1 – D1.4, Workshop | Import, place and arrange green screen elements |
| **CR17** | Overview of shots | 2,5 | Workshop | An 2D overview of all shots is required. |
| **CR18** | Timeline | 2,25 | Workshop D1.1, D1.2, D1.4 | The software requires a timeline to temporally arrange shots, sound and objects for animation. |
| **CR19** | Basic editing | 2/1,5/ 1,5 | Workshop, D1.1, D1.2 | Arranging, transitions, stretching, trimming of shots and sound on timeline. |
| **CR20** | Active / inactive shots | | Workshop | Ability to toggle shots to active and inactive mode. |
| **CR21** | Realistic light sources | 2,25 | Workshop, D1.2, D1.3 | Different light sources that represent real light setups like: spot lights, point lights, ambient light, sunshine are required. |
| **CR22** | Parameter editing for lights | 2,75 | Workshop | Ability to change parameters of light like: orientation, opening angle, intensity, colour. |
| **CR23** | Compatible with light libraries | 2 | Workshop | Open light libraries can be imported and used. |
| **CR24** | Different light effects | 2,75 | D1.1 – D1.4, Workshop | Integration of light effects like: projections, sunset, dawn, coloured lights. |
| **CR25** | Sound | 2 | D1.1, D1.2, D1.3, Workshop | Integration of sound files and arrangement on timeline is required. |
| **CR26** | Character posing | 2,5 | Workshop | Ability to apply predefined poses to rigged characters. |
| **CR27** | Pose library | 2,5 | Workshop | Software requires a set of predefined poses to import into the project. |
| **CR28** | Create, edit, retime basic animation | 2,75 | D1.1 – D1.4, Workshop | Ability to create basic keyframe animations, edit and change timing of existing keyframe animations. |
| **CR29** | Light animation | 2,5 | Workshop | Ability to create and apply basic animations to light sources. |
| **CR30** | Motion capture integration | 2,75 | D1.1, D1.3, Workshop | The integration of motion capture from motion suits. Motion recordings to animations. |
| **CR31** | Real-time simulation | 1,75 | D1.2, D1.4, Workshop | Physics simulation for cloth, soft bodies and fracture. |
| **CR32** | Art-directable simulation | 1,75 | Workshop | Ability to create simulations that follow predefined constrains e.g. paths, loops. |
| **CR33** | Integration of CGI elements | 2 | D1.1 – D1.4 | Ability to import and arrange CGI elements in image and scene view. |
| **CR34** | Real-time rendering | 2,5 | D1.1 – D1.4 | Rendering output of the software is required to be in real time. |
| **CR35** | Export to industry standard file types | 2,5 | D1.1 – D1.4 | Ability to export output and parts of the project to industry standard file types like: obj, stl, fbx, alembic, mp4, avi, csv, jpg, png. |
| **CR36** | Export to video / images | 2,25 | D1.1 – D1.4 | Ability to export the previsualized project to video file. |
| **CR37** | Standard hardware (standard PC, laptop or tablet) | 2,25 | D1.1, D1.2, D1.3, Workshop | Software has to run on standard PC, laptop, tablet hardware. |
| **CR38** | Organize work in projects | 3 | D1.1, D1.3 | Contain general information and all information on scenes, shots, layout, animations, cameras, effects, etc. |

  

| | | | | |
|---|---|---|---|---|
| **CR39** | Workflow / pipeline integration | 3 | D1.1, D1.4 | Integration into existing workflows, minimize training period, minimize work overhead, use previs for later production stages. |
| **CR40** | Intuitive interface (GUI / NUI) | 3 | D1.2, D1.3 | Clean and understandable user interface. |

### 2.5.2 Overlapping Requirements (OR)

The following requirements have been requested by at least two application partners. Requirements without any source have been added after workshop based on further discussion.

*Table 4: first.stage Overlapping Requirements*

| ID | Name | Priority | Source | Description |
|---|---|---|---|---|
| **OR1** | Access to asset library | 2,75 | D1.1, D1.2, D1.3 | Ability to connect to an asset library with a set of pre-defined assets to import into the project. |
| **OR2** | Create assets from sound / image files, sketches, websites | 2,25 | D1.1, D1.2 | Draft assets by defining them by images, sounds or websites. |
| **OR3** | VR layouting | 2 | D1.1, D1.4 | Use of virtual reality for the layout process. Benefit of better special presence in VR. |
| **OR4** | Rough to detailed set design | 2,5 | D1.2, D1.3 | Incremental set design that is not limited in its details or simplicity. |
| **OR5** | Camera position, angle | 2,5 | D1.1, D1.2, D1.4, Workshop | Ability add cameras to a scene and set individual position and orientation. |
| **OR6** | Rig cameras | 2,25 | D1.1, D1.2, D1.4, Workshop | Ability to import and add rig cameras to a scene. Camera rigs represent real camera setups like camera cranes etc. |
| **OR7** | Real camera setups | 1,5 | D1.1, D1.2, D1.4, Workshop | Use of real camera setups like shoulder cam, steady cam, camera cranes, etc. |
| **OR8** | Test lines of sight | 1,5 | D1.2, D1.3 | Ability to test if parts of the scene / stage can be seen from different viewpoints. |
| **OR9** | Camera motion | 2,5 | D1.1, D1.2, D1.4, Workshop | Ability to create and apply basic animations to cameras. |
| **OR10** | Aperture, focal length, lenses of cameras | 2,25 | D1.1, D1.2, D1.4, Workshop | Cameras feature parameters for Aperture, focal length, zoom and created by different lenses in real life. |
| **OR11** | Depth of field | 2,5 | D1.1, D1.2, Workshop | Cameras feature depth of field like real real cameras. |
| **OR12** | Physical information of camera | 1,5 | Workshop | Cameras store physical information of the real camera setup like mass and size. |
| **OR13** | Gesture control for camera | 1,25 | Workshop | Ability to control cameras using gestures. |
| **OR14** | Event driven / triggered sound | 1,5 | Workshop | Ability to automatically play sounds on defined events. |
| **OR15** | Multiple Tracks (sound) | 1,5 | Workshop | Ability to arrange multiple sounds in timeline with (partial) overlap. |
| **OR16** | Timeline editing (sound) | 1,5 | Workshop, D1.1 | Ability to position cut, trim, stretch sounds on timeline. |
| **OR17** | Humanoid character motion | 2,75 | D1.2, D1.3 | Ability to apply humanoid motions to rigged characters like walk, run, sit, etc. |
| **OR18** | Camera animation | 2,25 | D1.1, D1.2, D1.4, Workshop | Ability to create and apply animations for camera parameters to control zoom or depth of field over time. |

  

| | | | | |
|---|---|---|---|---|
| **OR19** | Optional photorealistic rendering | 1 | D1.2, D1.3, D1.4 | Ability to render in photorealistic quality if needed (not real time). |
| **OR20** | Export to lists | 1,75 | D1.1, D1.3 | List of assets with size etc. for cost calculation. |
| **OR21** | Share projects / collaboration | 2,25 | D1.1, D1.2, D1.3 | Share the project with several persons in a team. |
| **OR22** | Work on projects simultaneously | | D1.1, D1.2, D1.3 | Cloud-based synchronisation features. |
| **OR23** | Presentation mode | 2,25 | D1.3, D.14 | Special presentation mode to show previs to other people and customers to communicate ideas. |
| **OR24** | Physical interface | 1,25 | | Use of heavy cams, puppetry, etc. |
| **OR25** | Virtual Reality | 2 | D1.1, D1.4 | Use of VR for sketching, layouting, camera and animation. |
| **OR26** | Attach text and sound notes to objects, scenes, shots, etc. | 1,5 | D1.1, D1.2, D1.3 | Ability to make notes and attach them to objects, scenes, shots, etc. Notes can be made as text or voice recording. |

### 2.5.3 Application-specific Requirements (AR)

*Table 5: first.stage Application-specific Requirements*

| ID | Name | Priority | Source | Description |
|---|---|---|---|---|
| **AR1** | Import of stage model | 2,25 | D1.3 | Ability to import an accurate model of the theatre stage. Correct scale and measures. |
| **AR2** | Import of multiple scenes | 2,5 | D1.3 | Load multiple scene on the stage model with regard to its physical limitations. |
| **AR3** | 3D sketches | 1,75 | D1.1 | Directly sketch in 3D onto set / stage. Like Tilt Brush[1]. |
| **AR4** | Protected assets | 1,5 | D1.3 | Ability to make an asset protected, so it cannot be changed (position, rotation, scale) anymore. |
| **AR5** | Physical restrictions | 2 | D1.3 | Use physical restrictions of stage and machinery during layout, so no sets can be designed that violate the restrictions on a real stage. |
| **AR6** | Auto-update assets | 2,5 | D1.1 | Automatically update assets in the scene if the underlying asset file changed. |
| **AR7** | From shot to layout | 2 | D1.1 | Define shot first and create underlying scene based on that shot later. |
| **AR8** | Access shots from layout | 2 | D1.1 | Easy access to shot overview within the layout in 3D mode. |
| **AR9** | Asset search by meta data | 2,25 | D1.1 | Search for assets by meta data like size, colour, dates, etc. |
| **AR10** | Define shot by image | 1,5 | D1.1 | Creation of shot from image or collection of images that are arranged in 2D. |
| **AR11** | Shot planning | 2 | D1.2 | Ability to plan camera settings (position, orientation, lens, focal length etc.) and motion for each shot. |
| **AR12** | Position and scale viewfinder | 2 | D1.1 | Define camera settings (position, orientation, lens, focal length etc.) by making changes to the view finder. |
| **AR13** | Arrange camera according to shot | 2,5 | D1.1 | Define shot by images. Ability to arrange the camera in the scene to match the predefined shot. |
| **AR14** | Planning of real light sources | 2,25 | D1.3 | Ability to import or create light sources that exactly match real light sources. Restrict position and orientation to real conditions |
| **AR15** | Easy finding and applying of poses | 2,25 | D1.1 | Scrub through poses, preview, drag and drop of poses |
| **AR16** | Animation library | 2 | D1.1 | Software requires a set of predefined animations to import into the project. |
| **AR17** | Global timeline | 2,25 | D1.1 | Project requires a global timeline to arrange shots (can have local timelines) and sounds |

[1] https://www.tiltbrush.com/

  

| | | | | |
|---|---|---|---|---|
| **AR18** | Record character performance (mocap) | 1,75 | D1.3 | Directly apply motion capture to a rigged character and save as animation. |
| **AR19** | Special hardware for special cases | 1,5 | D1.3 | Integration of Motion suit, 3D scanner, tablets, virtual reality, gesture recognition for special tasks. |
| **AR20** | Light / music directing by motion performance | 1,75 | D1.3 | Ability to direct music / light by motion performance. |
| **AR21** | Simulation as substitution for standard VFX | 2 | D1.4 | Use of physics simulation instead of handcrafted visual effects to provide easy setup and compelling effects. |
| **AR22** | Intuitive, easy parameter adaption for VFX | 2,25 | D1.4 | Provide an easy and intuitive interface to set and change simulation parameters. |
| **AR23** | Simulation for stunt planning | 1,5 | D1.2 | Use physics simulation to plan visuals, setup and risks of stunts. Requires the ability to define specialized simulations. |
| **AR24** | Abstract rendering | 2,25 | D1.2 | Not too realistic rendering. Could harm creativity. |
| **AR25** | Export of assets | 2,5 | D1.4 | Export animations and other generated assets to use in later production stages |
| **AR26** | Compatible with industry standard protocols | 3 | D1.3 | Integration of other software/ hardware via protocols, e.g. light desk. |
| **AR27** | Layout directly on-stage model | 2,5 | D1.3 | Ability to layout a set directly on the theatre stage. |
| **AR28** | Simple timeline interface | | D1.1 | The interface to the timeline has to be simple so that none technical people can use it. In contrast to timelines in most professional applications. |
| **AR29** | Documentation | 2 | D1.4 | To be used by professionals the software requires a comprehensive documentation. |
| **AR30** | Licensing | 1,5 | D1.4 | To be used by professionals the software requires a industry standard license model. |
| **AR31** | Immersiveness | 2 | D1.4 | Presenting the audio-visual ideas of creatives in a immersive presentation mode. |
| **AR32** | Use from first draft to final rendering | 1,75 | D1.4 | Ability to use the software from first rough draft to very detailed final version of a project. Output is audio-visual (film / video). |
| **AR33** | Seamless change between devices | 1,5 | D1.2 | Ability to change between devices and continue working at the same point of the project as on the other device. |
| **AR34** | Switching between desktop and VR mode | 1,25 | D1.1 | Ability to change into VR mode when working on a suitable desktop PC with a VR headset plugged in. |
| **AR35** | Different levels of complexity | 2 | D1.3 | Software provides different levels of complexity from very easy for basic editing to more difficult for detailed and special tasks. |
| **AR36** | Specialized modules for different tasks | | D1.3 | Provide different modules for different tasks like: layout, light, camera, animation. Different modules differ in interface and interaction options. |
| **AR37** | Off the shelf input hardware | 1,5 | D1.3 | Only support hardware that can be openly bought and has no need for special setup that requires experts. |
| **AR38** | Project organisation in scenes and shots | | D1.1 | Organize the projects into scenes (represent the 3D arrangement of objects, lights, cameras, animations) and shots (represent a particular view from a camera onto the scene). |
| **AR39** | Light projections | | D1.3 | Ability to create special lights that project images onto the set / stage. |

  

## 2.6 Hardware Requirements

Based on the classified requirements listed above, hardware requirements can be derived. In the common requirements section, two requirements relate to hardware requirements definition. Requirement **CR34 (Real-time rendering)** defines the needs for graphical rendering capabilities of the hardware. **CR37 (Standard hardware: PC, laptop or tablet)** defines the needs for ergonomics, acquisition, price and computational power. Standard PC and Laptop already satisfy requirement **CR34** as they are capable of real time rendering with sufficient quality. The project decided on using Unity[2] as a rendering engine. It satisfies all requirements listed above, supporting Windows, Mac, iOS and Android operating systems. Therefore, all PC, laptops and tablets can run the software. Specialized system requirements can be made from Unity´s hardware requirements:

**PC and Laptops Hardware Requirements**

2 GHz Dual Core, 4 GB RAM, Graphics card with DX9 (Shader model 3.0) or DX11 with Feature-Level 9.3-Capability.

**Tablet Hardware Requirements**

iOS 7.0 or higher, Android: OS 2.3.1 or higher. ARMv7 (Cortex) CPU with NEON-Support or Atom-CPU, OpenGL ES 2.0 or higher.

In order to use virtual reality, satisfying requirement **OR25 (Virtual Reality)**, special hardware requirements have to be met. The project decided on using the HTC Vive[3] which features room scale VR with best hand tracking available. In order to support the Vive the following requirements can be defined.

The use of VR will be optional in the first.stage software. The listed requirements for VR only have to be satisfied if the user wants to enter the VR mode of the software:

**HTC Vive Hardware Requirements**

Intel Core i5-4590 or AMD FX 8350 (equivalent or better), 4 GB RAM, NVIDIA GeForce GTX 1060 or AMD Radeon RX 480 (equivalent or better).

# 3 Functionalities

For a better understanding of how the core functionalities will be implemented, we first present a list of all demonstrators of the project (Table 6). In the following sections, we then present the core functionalities and match them to the individual demonstrators.

## 3.1 List of Demonstrators

*Table 6: List of demonstrators and their functionalities (UniHB for University of Bremen, MoSt for MovieStorm)*

| Deliverable | Title | Responsible | Dissemination Level | Due in Month |
|---|---|---|---|---|
| **D2.1** | Demonstrator for Asset Creation | NL | Public | 14 |
| Simple demonstrator for evaluation of features. Browsing and selection in asset libraries, scanning of simple real- world assets and simple physical simulations will be supported. | | | | |
| **D2.2** | Final Demonstrator for Asset Creation | MoSt | Public | 24 |

[2] https://unity3d.com
[3] https://www.vive.com

  

| | | | | |
|---|---|---|---|---|
| Final demonstrator for project-based evaluation. In addition to refined functionality already supported by D2.1, the final demonstrator is able to scan complex 3D objects, interfaces to at least 3 hardware sensors, offers complex content creation by physical simulation and supports digital crafting. Advanced NUI interaction is supported. | | | | |
| **D3.1** | Demonstrator (Mock-Up) | UniHB | Public | 14 |
| Simple demonstrator for evaluation of NUI interaction featuring basic scene (and shot) layout, as well as basic character, effect and camera animation and simple editing functionality. | | | | |
| **D3.2** | 2nd Demonstrator | UniHB | Public | 24 |
| Demonstrator supporting precise layout, advanced animation (including camera planning) and physics simulation. Support of different hardware for motion recognition. | | | | |
| **D3.3** | Final Demonstrator | UniHB | Public | 35 |
| Final demonstrator supporting all operation based on NUI interaction. All results from the evaluation are incorporated. | | | | |
| **D4.2** | Animation Engine Demonstrator | MoSt | Confidential | 18 |
| Interface to animation engine(s). | | | | |
| **D4.4** | Sensor Hardware Research Demonstrators | Rokoko | Confidential | 24 |
| A set of approximately 20 motion tracking suits by RoKo. | | | | |
| **D5.2** | Modelling Workflow within Demonstrator | YoMo | Public | 33 |
| Demonstrator connecting different previs tools through a pipeline. | | | | |

Based on the requirements analysis from Section 2, in the following core functionalities and additional functionalities are derived in the following. While core functionalities are a definite must, additional functionalities might be researched and implemented in the end of the project. The classification into core and additional functionalities is subject to be adapted based on the iterative and agile design and development process of the first.stage project.

The requirements are chosen to fulfil one or more requirements listed in the tables from categorised requirements in Section 2.5. In most cases more than one requirement can be satisfied by the functionality. In addition to a detailed description, the expected implementation in a demonstrator from the table of demonstrators in Section 3.1 is listed. The functionality description features the average priority of the related requirements chosen by all partners. The priority is displayed by (pX) where X is the average priority from Section 2.4. The assignment of requirements to its functionality was done using the card sorting method [3].

## 3.2 Core Functionalities

**CF1: Project Structure**

**Related requirements:** CR38, AR38

**Implementation in demonstrator:** D3.2 / D3.3

**Description:** All application areas described their work as project based. Therefore the base structure to organize work and store data in the first.stage software will be “projects” (p3). A project contains basic information such as: Project name, customer, assigned persons, creation and edit dates as well as deadline or publishing dates. Furthermore, all data related to the

  

project such as: 3D models, scene graph, textures, animation data, shots, renders videos, images and lists and meta data are stored within the project.

Projects are organized into two sub structures. “Scenes” contain all 3D related information like the position of all 3D models stored in a scene graph, textures assigned to objects, positions of lights and cameras as well as animation data. “Shots” define a view onto a scene. They are represented as 2D views and have a camera from the scene assigned defining this specific viewpoint.

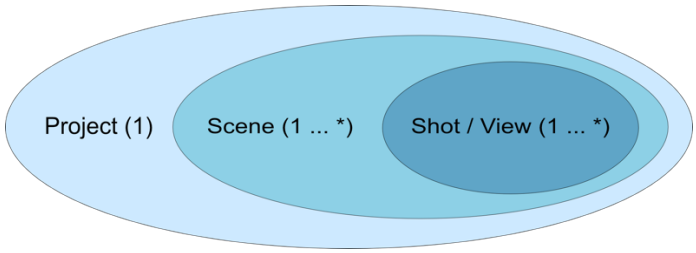


*Figure 5: Internal structure of the first.stage software for storing and presenting data*

Even though the related requirements are only overlapping and application specific requirements, all partners implicitly referred to their work as projects. They also made the described distinction between scenes and shots or viewpoints. Therefore, this structure was chosen as a core concept that applies to all application areas.

**CF2: Import**

**Related requirements:** CR1, CR2, CR3, CR4, CR5, CR6, CR16, CR23, CR33, CR39, AR1

**Implementation in demonstrator:** D3.2

**Description:** The software will support import functionality from various file types. This was mentioned as a key functionality required by the software. The import functionality will support industry standards used in the application areas (p3). For the import of 3D models, the software will support OBJ, STL and FBX file types (p3). This ensures integration into the workflow with standard 3D modelling software like Maya, CAD and Cinema 4D (p3). For image and texture import BMP, TIF, TGA, JPG, PNG and PSD file types are supported. Sound files can be imported as MP3, OGG or WAV (p1.75). Complete characters that contain a 3D model and textures as well as rigged objects can be imported with as Unity prefab objects (p2). Green screen (p1.5) and CGI (p2) elements can be imported as rendered videos with the file types: MOV, MPG, MPEG, MP4, AVI, ASF.

The import functionality from the described file types ensures great integration into existing workflows and pipelines within the application areas (p3). For example the theatre stage model that is already existent in CAD applications can directly be imported into the first.stage software and used as an asset (p2.25).

**CF3: Shot Management**

**Related requirements:** CR17, CR20, AR7, AR10

**Implementation in demonstrator:** D3.2

  

**Description:** The first.stage software will provide an overview of all shots (p2.5). The order of shots can be changed, and individual shots can be set active or inactive (p2). New shots can be added by selecting or adding a camera to the scene. Furthermore, a shot can be defined by sketches or images (p2). In this case the layout for the underlying scene can be started using the shot as background (p1.5).

**CF4: Sketching/Modelling**

**Related requirements:** CR7, OR2, OR4, AR3, AR35

**Implementation in demonstrator:** D3.1 / D3.2

**Description:** The software will provide functionality to draw sketches in 2D (p2) when using a touch device or in 3D when using VR (p1.75). If sketches were made outside of the first.stage software they can be imported as images. Sketches as well as images, sound files or websites can be used to create first assets for a scene or shot (p2.25). This enables creative with little technical knowledge to visualize and communicate their ideas. This setup stage has a low level of complexity and can be used by every user independent of their technical knowledge (p2). A rough set design can be achieved in this stage.

**CF5: Assets**

**Related requirements:** CR9, OR1, AR4, AR9

**Implementation in demonstrator:** D3.1

**Description:** In addition to the ability to import assets to a project, the software will provide the functionality to connect to an asset library (p2.75) which has a set of predefined assets to use. Imported assets as well as assets from the database can be searched by meta data like size and colour to find the assets needed (p2.25). In addition, placeholder assets can be created (p2.25). They are defined by a name which describes the object that will later replace them. Optionally the size can be defined for better representation of the later object within the scene. All assets can be made protected (p1.5). Properties of protected assets cannot be changed anymore.

**CF6: Layout**

**Related requirements:** CR8, CR10, CR11, CR12, CR13, OR3, OR4, AR7, AR27

**Implementation in demonstrator:** D3.1

**Description:** In order to design a set or stage, the software will provide a layout mode. Assets can be instantiated, and their position, rotation and scale can be defined (p3). Instantiated assets can be grouped / parented (p2.5). Grouped assets form a new asset that can be positioned in the scene and all grouped assets inherit the parameters of the group.

Placing assets in the scene is assisted by a snapping tool (p2) that helps aligning objects. The layouting can be done in different views. The software provides a standard 3D view on the scene (p2.75). In addition, a top down 2D view is provided (p2.25). With the use of VR, a virtual in-scene view is provided that gives improved spatial awareness and immersiveness (p2).

  

With the described layout mode, a rough to detailed set design can be achieved (p2.5). When working with fixed assets like a theatre stage it can be placed first and made protected, so further layout can be done on stage (p2.5) without changing the fixed stage. Having done sketches of shots first, these can be used a background for the layout phase to build a set / stage according to the drafts (p2).

**CF7: Camera Control**

**Related requirements:** OR5, OR6, OR7, OR8, OR10, OR11, OR12, OR13, AR11, AR12, AR13

**Implementation in demonstrator:** D3.1

**Description:** The first.stage software will provide the ability to instantiate one or more cameras in a scene and set their position and rotation (p2.5). Internal camera settings like aperture, focal length (p2.25) depth of field (p2.5) and lenses can be modified. Positioning and parameter modification can either be done by a classic 2D interface or using gestures in VR. Gesture control (p1.25) supports direct manipulation of the camera viewfinder. Each camera can visualize its view inside the 3D scene in order to test the lines of sight (p1.5).

Cameras can be imported as rigged cameras (p2.25) to support real physical camera setups like camera cranes. Other camera setups used in real life like shoulder and steady cams are also supported by the software (p1.5). Using such camera setups, the physical information (p1.5) of this setup are stored and can be exported as a list.

As each shot is represented by a camera, it is possible to plan shots by viewing previews of the camera image and define them as shots (p2). Having sketched a shot beforehand, it is also possible to set up the camera according to the shot by projecting the sketch into the scene (p2.5).

**CF8: Camera Motion**

**Related requirements:** OR9, OR18, OR24

**Implementation in demonstrator: D3.1**

**Description:** In addition to static camera control the software will support camera animation (p2.5). Camera position and rotation as well as internal parameter can be changed over time using key frame animation and performance recording (p2.25). To account for real cameras used by camera men in the production stage later on, a physical interface (p1.25) with tracked heavy cameras can be used during performance recordings.

**CF9: Light**

**Related requirements:** CR21, CR22, CR23, CR24, CR29

**Implementation in demonstrator:** D3.1

**Description:** Lights sources can be placed in the scene, setting their position, rotation and scale. The additional internal parameters light brightness, opening angle and colour can be modified (p2.75). The software provides a set of predefined lights that match realistic light sources from real life (p2.25). Furthermore, lights can be imported from open light libraries

  

(p2). Motion of light can be achieved using key-frame animation on the light object (p2.5). Special light effects like sunset, dawn, indoor and projections will be supported (p2.75).

**CF10: Posing**

**Related requirements:** CR26, CR27, AR15

**Implementation in demonstrator:** D3.1 / D3.2

**Description:** The software provides the functionality to apply poses to rigged characters (p2.5). Poses can be chosen from a predefined set of poses from a library (p2.5). Searching and applying poses to characters can be done with a quick and easy to use interface (p2.25).

**CF11: Animation**

**Related requirements:** CR18, CR28, CR29, OR18, AR16

**Implementation in demonstrator:** D3.1, D4.2

**Description:** The first.stage software provides functionality to create basic key-frame animations (p2.75). Existing animations can be edited by changing key-frame properties and whole animations can be retimed changing the time between key-frames (p2.75). Animations can be applied to rigged characters, assets, lights and cameras (p2.5 / p2.25).

Animations can be created by the user or chosen from a predefined set of animations from a library. All animations can be arranged on local scene or shot timelines (p2.25).

**CF12: Motion Capture**

**Related requirements:** CR30, OR17, AR18, AR19

**Implementation in demonstrator:** D3.2, D4.4

**Description:** Motion Capture functionality will be integrated in the first.stage software (p2.75). This includes the functionality to connect different motion capture hardware to the system as well as record and save human performances as animations. This allows to use real humanoid motion (p2.75) to animate characters within the first.stage software.

Recording a human performance (p1.75) is a special application which may only be needed to a small extend in the process of a project. Therefore, is uses special hardware (p1.5) like the SmartSuite Pro[4] which only needs to be connected for this task.

Setup and recording with supported motion capture hardware will be easy and will not require special technical knowledge.

**CF13: Visual Effects**

**Related requirements:** CR31, CR32, AR21, AR22

**Implementation in demonstrator:** D3.2

**Description:** The first.stage software will support the functionality to create, edit and arrange visual effects (VFX). As visual effects often replicate physical behaviour or use physics

[4] https://www.rokoko.com/en/smartsuit-pro

  

simulation, the first.stage software will use real time physics simulation (p1.75) to create visual effects and substitute the complex process (p2) of handcrafted visual effects. This will be achieved using art-directable simulation (p1.75) by providing an intuitive and easy to use interface for setting and editing simulation parameters. This interface relies on capturing human performance and translating it to simulation parameters like capturing path and timing of a performance (p2.25).

**CF14: Export**

**Related requirements:** CR35, CR36, OR20, AR25

**Implementation in demonstrator:** D3.3

**Description:** The first.stage software will provide export functionality to industry standard file types (p2.5). The export functionality can be divided into categories. The export to video and images (p2.25) will support MOV, MPEG, MP4 and AVI as well as JPG and PNG file formats. Assets created within the first.stage software like animations, motion capture and visual effects as well as the created set can be exported to FBX files (p2.5). Furthermore, the software will support the export to lists (p1.75). Lists contain data about the project or scenes. This can be all shots and scenes with timing information or all assets and their dimensions that are used in the project or particular scene. Lists will be exported in CSV file format.

**CF15: Interfaces**

**Related requirements**: CR40, OR24, AR28, AR22, AR36

**Implementation in demonstrator**: D3.1

**Description**: The first.stage software will provide an intuitive interface (p3) for both GUI (graphical user interface) on a desktop / laptop computer and tablet as well as an intuitive NUI interface using motion tracking in VR.

Intuitive and easy to use interfaces will include a simple timeline interface which in other applications is normally very complex and not usable for non-technical persons. Also, the creation of physics simulations for visual effects can be very complex. This will be solved by a simple performance interface for simulation parameter editing (p2,25). To foster the imagination but also show restrictions physical interfaces (p1,25) like heavy cameras will be supported by the first.stage software.

The intuitive interface will feature specialized modules for different tasks. Each module will feature a specialized interface supporting the exact interactions possible for the related task (high priority, added after the requirement periodisation process).

**CF16: Devices**

**Related requirements:** CR37, OR25, AR37

**Implementation in demonstrator:** D3.1

**Description:** The first.stage software will support Windows and Mac computers with standard desktop and laptop hardware (p2.25). In addition, a tablet version of the software for iOS and

  

Android will be available later. The software will also support VR devices (p2), starting with the HTC Vive when using a Windows computer with sufficient processing hardware.

All devices required to use the software will be “off the shelf” hardware that is openly available for purchase (p1.5).

**CF17: Seamless Change between Interface and Devices**

**Related requirements:** CF15, CF16, AR33

**Implementation in demonstrator:** D3.3

**Description:** Although the seamless change between devices itself was not highly prioritised by the application partners (p1.5), the C15 and CF16 imply this as a core functionality. This requires a constant synchronisation of the projects, ideally over a cloud-based service. Additional architecture development and infrastructure will be needed.

**CF18: Virtual Reality**

**Related requirements:** CR10, CR40, OR3, OR25, AR3, AR34

**Implementation in demonstrator:** D3.1

**Description:** The first.stage software will support virtual reality devices like the HTC Vive. Virtual reality (p2) provides an immersive (p2) way of displaying 3D scenes with improved spatial awareness and accuracy compared to standard monitors. The tracking features integrated into current VR systems also provide an accurate, easy to set up, and cheap solution for hand tracking. These features will provide the first.stage software with the functionality to do 3D sketches (p1.75) and layouting (p2) directly in the 3D scene rather than on a 2D screen. The tracking features will provide an intuitive NUI interface (p3) for these tasks. In order to support the workflow with VR it will be possible to switch seamlessly between desktop and VR mode (p1.25).

Some interviewees had concerns regarding the comfort of VR headsets (D1.2). However, as the currently available commercial solutions are first generation, future headsets are expected to be way more comfortable.

In addition to the requirement of VR based on the user’s point of view in this deliverable, in the following NUI concept (D1.6) we will argue for VR from a interaction perspective.

**CF19: Collaborative Work**

**Related requirements:** OR21, OR22, OR29, AR36

**Implementation in demonstrator:** D3.3

**Description:** In order to account for the project-oriented work with multiple people the software will provide collaboration functionality (p2.25). This includes the ability to save projects and share them with other persons. Later versions of the software will also support simultaneously work of multiple people on the same projects in the cloud.

  

Supporting the communication between project members they can attach text or sound notifications (p1.5) to all objects, scenes and shots. New or changed notifications will be highlighted.

In order to provide an intuitive and easy to use interface for all users with diverse creative and technical skills, the software will provide specialized modules for different tasks featuring a specialized interface tailored to the related task.

**CF20: Presentation**

**Related requirements:** OR23

**Implementation in demonstrator:** D3.3

**Description:** The software will include an interactive Presentation mode (p2.25) in order to present the previsualized project or specific parts to others.

## 3.3 Additional Functionalities

**AF1: Physical Layout**

**Related requirements:** CR14, CR15, CR31

**Implementation in demonstrator:** D3.1

**Description:** In order to account for physical restrictions and measures during the layout a physical layout mode is provided (p2). In this mode objects follow real life physical behaviour using a real-time physics simulation (p1.75). Objects fall down and collide. They cannot be placed intersecting. In addition, measures like distances between objects or size are displayed within the 3D scene (p2.25).

**AF2: Editing**

**Related requirements:** CR18, CR19, OR15, OR16, AR17, AR28

**Implementation in demonstrator:** D3.2 / D3.3

**Description:** The software will provide basic editing functionality (p2). This includes the ability to arrange temporal objects like shots, sound and animations on a timeline (p2.25). A global timeline (p2.25) for the whole project as well as local timelines for each scene and shot are provided. Editing objects on a timeline supports simple cut, stretch and repeat functions (p1.5). The timelines feature multiple tracks (p1.5) to arrange multiple shots or sounds.

The timeline provides a simple interface that can be used by non-professionals.

**AF3: Sound**

**Related requirements:** CR25, OR15, OR16

**Implementation in demonstrator:** D3.2 / D3.3

**Description:** Sound can be imported into the project (p2) and placed in the timeline featuring multiple tracks (p1.5) and simple editing tool (p1.5) like cut, stretch, loop and mute. Sounds will automatically play according to their position in the timeline.

  

**AF4: Rendering**

**Related requirements:** CR34

**Implementation in demonstrator:** D3.1

**Description:** The first.stage software will support real-time rendering in a sufficient rendering quality on all supported devices (p2.5).

## 3.4 Application-specific Functionalities

Even though many of the requirements from the tables in Section 3.5 could be assigned to the core functionalities, there are application-specific requirements that cannot be accomplished due to their very specific nature. This section provides a list of all requirements that cannot be fulfilled by demonstrators of the first.stage project with detailed descriptions why they could not be included into the core functionalities.

### 3.4.1 Animation

**AR8 (Access shots from layout):** As animation production is very focused on shots they require a continuous reference to shots. The software will already feature the ability to use shot sketches for building 3D scenes. But a continues reference to all shots during layout will no be provided. In order to provide a simple and streamlined workflow, the proposed structure of separated scene layout in 3D and shot overview in 2D will be used.

### 3.4.2 Film / Commercial

**OR19 (Optional photorealistic rendering):** The ability to use optional none real time rendering in photorealistic quality was not added to the core functionality due to its low prioritisation by all application partners.

**AR23 (Simulation for stunt planning):** Planning the visual, execution and risks of stunts is required by in the film/commercial application area. But planning these factors in detail using physics simulation requires a precise physics simulation that can´t be calculated in real-time as well as a complex interface to control all parameters. Therefore it is not feasible in the first.stage software to be implemented.

**AR24 (Abstract rendering):** Realistic rendering with detailed visual information could harm the creativity of persons using the software. A dedicated rendering mode for abstract rendering leaving out details needs additional effort that can´t be provided during development of the first.stage software. But a low detailed rendering can be achieved using low detail assets during the layout phase.

### 3.4.3 Theatre / Stage

**AR2 (Import of multiple scenes):** For stage production it is required to plan with multiple scenes on one stage in order to plan storage space and the ability to change the sets. The import of multiple scenes onto one stage model is a very specific requirement only listed be the theatre. Is can be achieved using the core functionalities by directly building a scene, integrating multiple sets.

**AR5 (Physical restrictions):** Planning with physical restrictions like the suspension length or turning angles and velocities of the theatre stage is required for stage productions. But integrating these restrictions is only requested by this application partner and would need significantly more implementation effort which can´t be justified.

  

**AR14 (Planning of real light sources):** Having virtual representations of real lights used in the theatre is required for stage productions. The software will already support the ability to modify light parameters to match real light sources. But implementing the exact measures for various light sources would need extensive time that is not scheduled for this project.

**AR26 (Compatible with industry standard protocols):** Supporting protocols for light desks or motor control that are used by theatres requires the implementation of different low-level communication protocols that are very specific to the theatre application. It would need extensive time that is not scheduled for this project.

**AR39 (Light Projections):** Using light projections, replicating the effect of projectors, is required by theatre productions. This requirement would add additional complexity to the light setup and is currently only required by one application area. It could be added at a later production stage of the software.

### 3.4.4 Special Effects

**AR29 (Documentation):** The requirement of a comprehensive documentation applies for a final product that will be sold, but not for the demonstrator-based research project.

**AR30 (Licensing):** The requirement of an industry standard licensing model applies to a final product that will be sold. The outcome of this project will be demonstrators that will not form a final product that will be sold.

**AR32 (Use from first draft to final rendering):** The software should focus on the previsualisation part of a production rather than the whole production process. Application partners agreed that this will not include final rendering. Medium quality in rendering and animations are sufficient to visualize the idea and intention of the production.

# 4 Conclusion

In this document, we present a comprehensive analysis of the requirements for previsualisation and derive a set of core functionalities for the demonstrators of the first.stage project.

The requirements analysis presents the results of the requirement reports from deliverables D1.1 to D1.4 and the results from a workshop with all partners. A prioritisation of all gathered requirements was done by all partners and is presented. In addition, all requirements are categorized into common, overlapping and application specific requirements.

Based on the requirements analysis the core functionalities are defined. 22 core functionalities could be defined referencing back to the requirements. Each functionality is assigned to a demonstrator indicating its planned implementation. Requirements that will not be implemented within the demonstrators are listed separately.

This document defines the core functionalities for the previsualisation system of the first.stage project and therefore finalizes milestone MS2 – Core functionalities defined. It will further define the base for all future work of the project and will have major influence on the definition of the NUI concept in D1.6, the system architecture in D4.1, sensor hardware concept in D4.3 as well as the implementation

  

of all demonstrators (D2.1, D3.1, D3.2, D3.3, D4.2, D4.4) that will be developed in the first.stage project.

# 5 References

[1] Murali Chemuturi. Requirements Engineering and Management for Software Development Projects. Springer Publishing, 2013.

[2] Dai Clegg and Richard Barker. Case Method Fast-Track: A RAD Approach. Addison-Wesley Longman Publishing, 1994.

[3] Donna Spencer. Card Sorting: Designing Usable Categories. Rosenfeld Media, 2009.

# 6 Document History

| Ver. | Date | Changes | Author |
|---|---|---|---|
| V1.0 | 30.11.2016 | Initial draft | Thomas Muender |
| V1.1 | 03.01.2017 | Added list of demonstrators | Thomas Fröhlich |
| V1.2 | 30.01.2017 | Added list of all requirements and added core functionalities | Thomas Muender |
| V1.3 | 01.02.2017 | Added periodisation | Thomas Fröhlich |
| V1.4 | 04.02.2017 | Added descriptions | Thomas Fröhlich |
| V1.5 | 07.02.2017 | Final improvements | Thomas Fröhlich |
| V1.6 | 09.02.2017 | Proofreading and final formatting | Peter Knackfuß |
| V1.7 | 20.02.2017 | Final edit | Thomas Münder |
| V2.0 | 29.05.0218 | Revision requirements and functionalities | Dirk Wenig |

  