# Optimizing User-centered Design for CreateData4AI

Mathis Hoffmann

03 November 2025

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

# Background | Manual Annotation of Text Data is Challenging

### The Situation

- Zettabytes of data are generated every day
- 80% of this data is unstructured text

→ How to make text useful for AI applications?

### The Challenge

- Training AI models requires annotated data
- Annotation eats up most of the time – "80/20 Rule"

→ Manual annotations is costly, inefficient, not scalable!

### CreateData4AI (CD4AI) – The Solution?

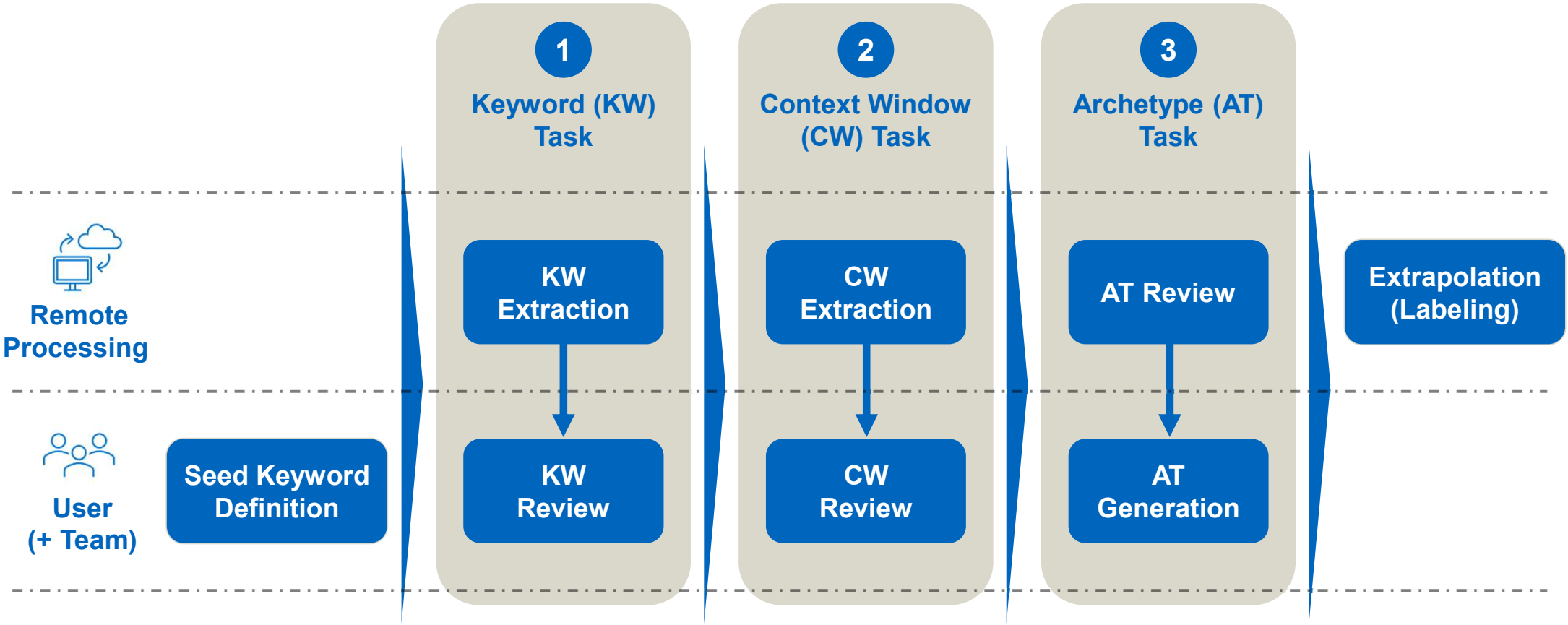
- **What?** – A human-in-the-loop system to automate labeling of text corpora
- **Goal?** – Drastically reducing manual effort in the annotation process
- **How?** – Combining human domain expertise with Natural Language Processing algorithms

"**Frontend**" web application to interact with the workflow

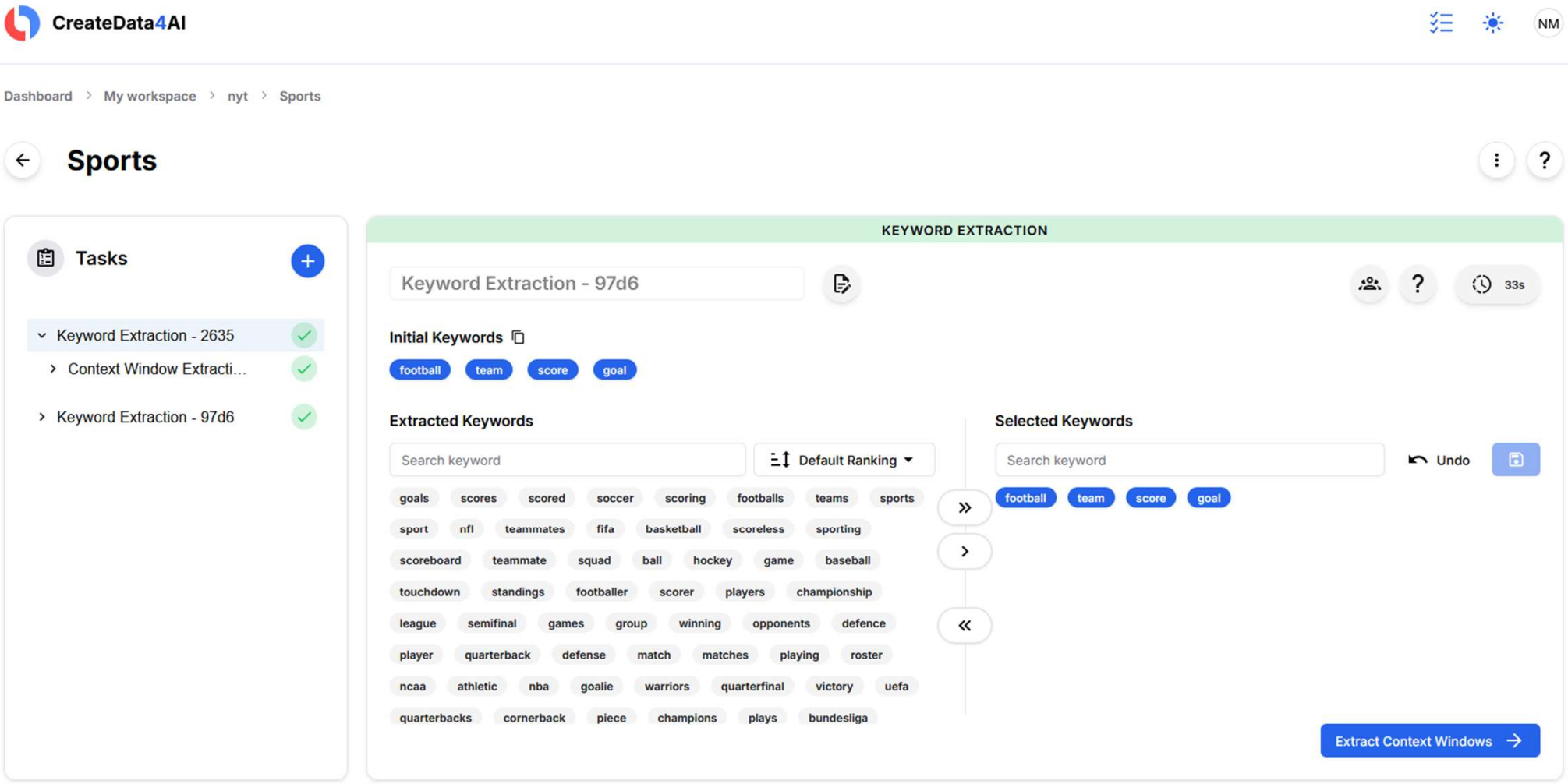
"**Backend**" NLP engine to process and label text data

# Background | CD4AI's Hybrid Human-Algorithm Approach

CD4AI is a **task-oriented** web app, with **human-algorithm** and **human-human co-creation** at its core.



# Background | The CD4AI Web Application



# Background | Open Issues with the CD4AI Prototype

## User-centered Design

Former research has focused on the CD4AI framework. Usability research on the web application has not taken place.

## Feature Maturity

Key Parts and recent advancements of the CD4AI have not been implemented or are not functionally usable.

## Public Accessibility

The web application is not publicly accessible. Independent usage is not possible without external support and insider knowledge.

# Research Questions

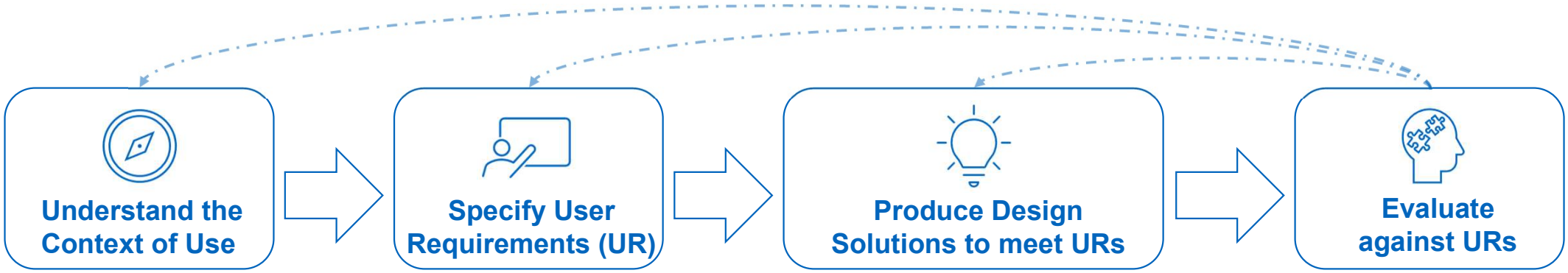
**RQ1:** What efforts can be taken to reduce manual effort in the CD4AI web application?

**RQ2:** How can the CD4AI web application be iteratively optimized for usability?

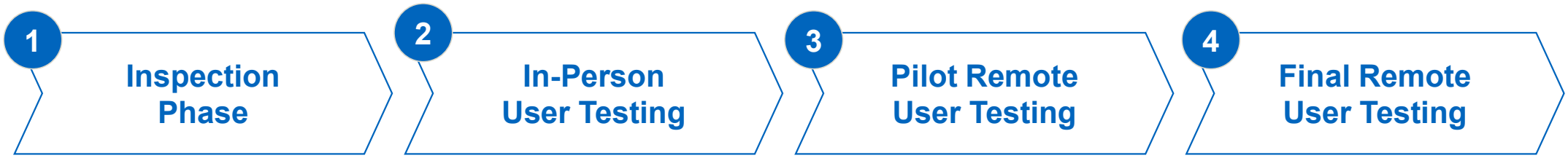
**RQ3:** What technical adaptations are required to transform the CD4AI application from a research prototype into an externally usable system?

# Methodology | Iterative User-centered Design[1] (ISO 9241-210)

**User-centered Design** is an **iterative framework** to advance low-fi prototypes to **mature systems**.




**4 iterations** of UCD were conducted, each with a **specific purpose** and **dedicated methods**.




*1) ISO 9241-210:2019 – Ergonomics of human-system interaction — Part 210: Human-centered design for interactive systems*

# Results | Inspection Phase (Phase 1)

**In phase 1**, the prototype was inspected and improved **without user intrusion**.

## Heuristic Evaluation

Evaluate the interface against established usability principles (heuristics).

## Cognitive Walkthrough

Step through tasks, predicting user thought processes.

## Action Analysis

Break tasks into detailed user actions to assess efficiency and workload.

**→ Nielsen's 10 Usability Heuristics**

served as the central guidelines for heuristic evaluation, and for validation of ideated features.

# Results | Inspection Phase (Phase 1)

Initial **inspection and ideation** informed the implementation of **over 28 key adaptations**.

## Administration (3)

- Email Notification System
- Project-Level Task Dashboard
- …

## Task Interaction (8)

- Task Status Banners
- Auto-Confirm Seed Keywords
- …

## Team Collaboration (2)

- Auto-Merge Selection Tasks
- Selection Tasks Consistency

## Pipeline Finalization (2)

- LLM-based Archetype Generation
- Archetype-driven Document Classification

## Deployment (3)

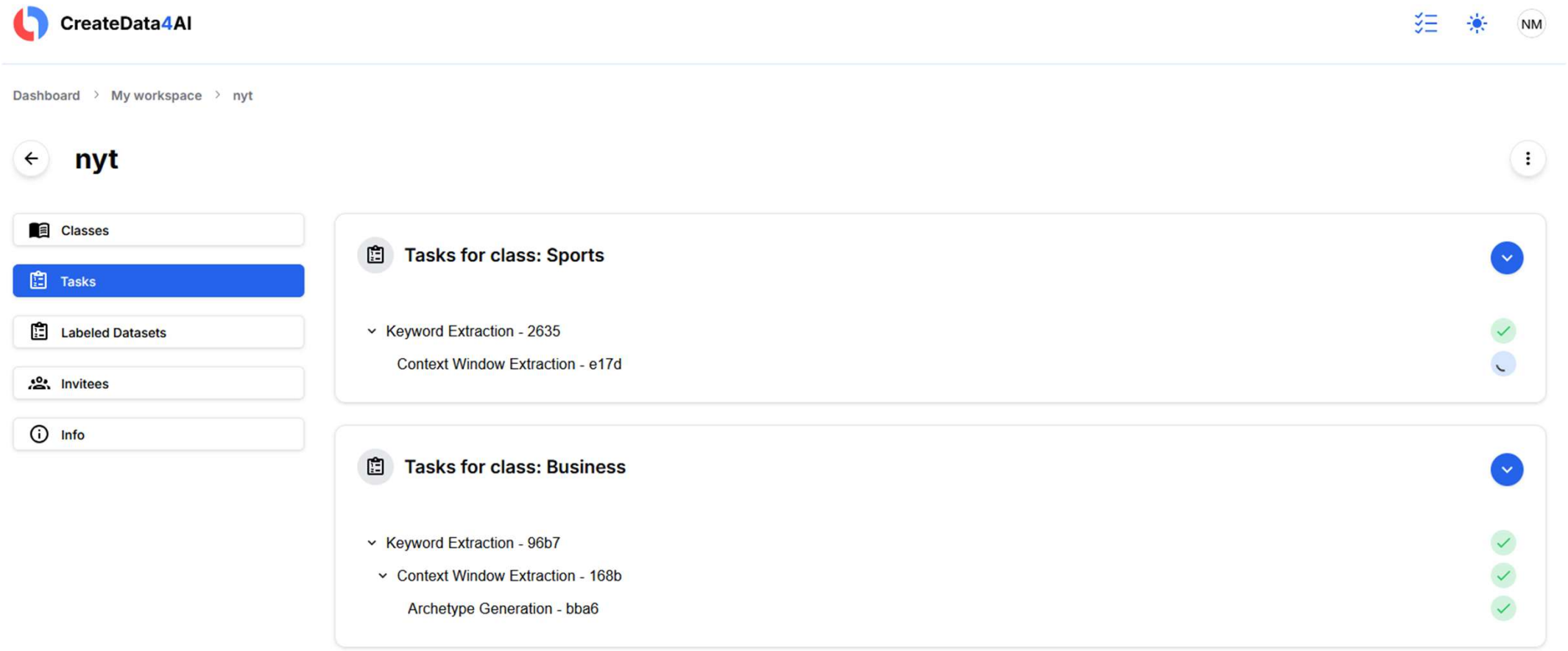
- Public Accessibility: https://createdata4ai.com
- GPU Acceleration
- …

## Bug Fixes (10)

- Search Bars
- Task Cancellation
- …

# Example: Task Dashboard

A task dashboard makes it significantly easier to **track task statuses** in a unified manner.



# Results | In-Person User Testing (Phase 2)

**In phase 2**, eight test users were observed completing an exemplary task while “**thinking aloud**”.

## 1 Voluntary Recruitment

- 8 Researchers (SEBIS Chair)
- 100% Technical Background
- Voluntary Participation

## 2 Test Task Completion

- In-person 1-on-1 sessions
- Guided & observed
- NYT news article classification (600 samples, 3 classes)

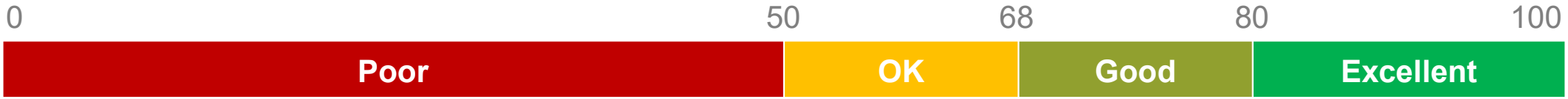
## 3 Follow-up Survey

- **System Usability Scale**
- Free-text questions

---

## The System Usability Scale (SUS)

- Popular, technology-agnostic usability research instrument
- 10 item survey (5-scale Likert)
- Yields a score from 1-100



# Results | In-Person User Testing (Phase 2)

**SUS Score**

**75.6**

*indicates*
***GOOD***
*usability*

The identified usability issues informed the design and implementation of **16 further enhancements**.

### Task interaction (3)

- Redesigned Tab Components
- Keyword Review Enhancements
- …

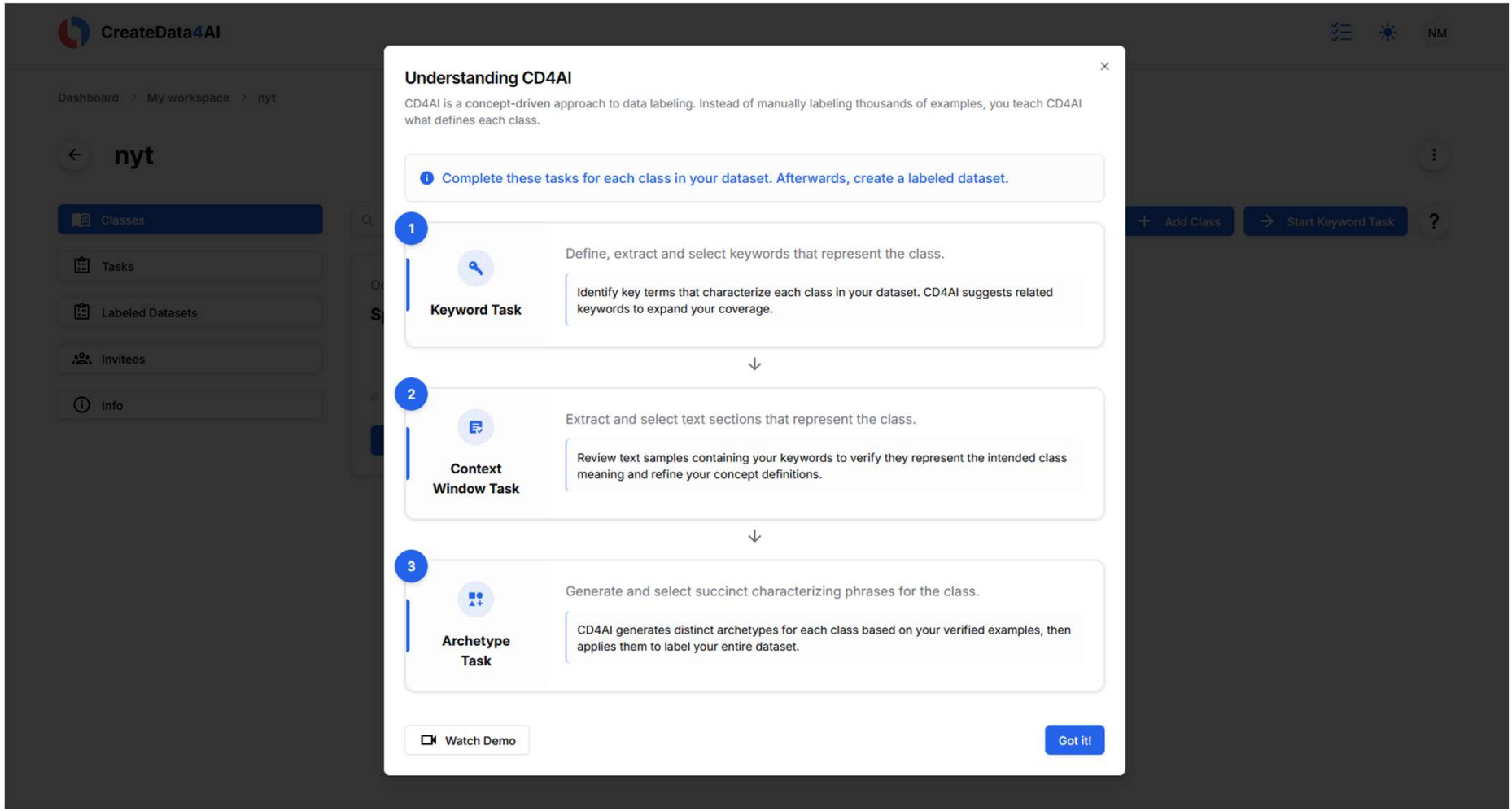
### Guidance & Help (3)

- Onboarding Wizard
- Simplified Terminology
- …

### Other (10)

- Consistent In-App Notifications
- Visual Consistency
- …

# Example: Onboarding Guide



# Results | Pilot Remote User Testing (Phase 3)

**In phase 3**, remote test users completed an exemplary task **without guidance**.

## 1 Recruitment via Prolific

- 5 Strangers (Prolific)
- 60% Technical Background
- Paid Participation

## 2 Test Task Completion

- NYT news article classification (400 samples, 2 classes)

## 3 Follow-up Survey

- System Usability Scale
- Feature-related questions
- Free-text questions

# Results | Pilot Remote User Testing (Phase 3)

**In phase 3**, remote test users completed an exemplary task **without guidance**.

**SUS Score**

63.0

↘ **- 12.6**
*indicates OK usability*

**Task Success Rate**

80%

*4 out of 5*

**Task Completion Time**

29.5 min

*vs. 11.5 min "expert" TCT*

## What was the issue?

- Users felt **overwhelmed** and **insecure** with the task.
- Guidance provided by the in-app onboarding wizard was **insufficient** to let users confidently perform the task.

*"There wasn't really a clear overall explanation of what and how the system worked […]. With no experience prior to this, it is hard to really tell if what I was doing was right."*
– Participant 3.5

# Results | Pilot Remote User Testing (Phase 3)

Based on the findings, further improvements to the **onboarding experience** were made.

### Demonstration Videos

Guide users through the CD4AI workflow with detailed instructions.

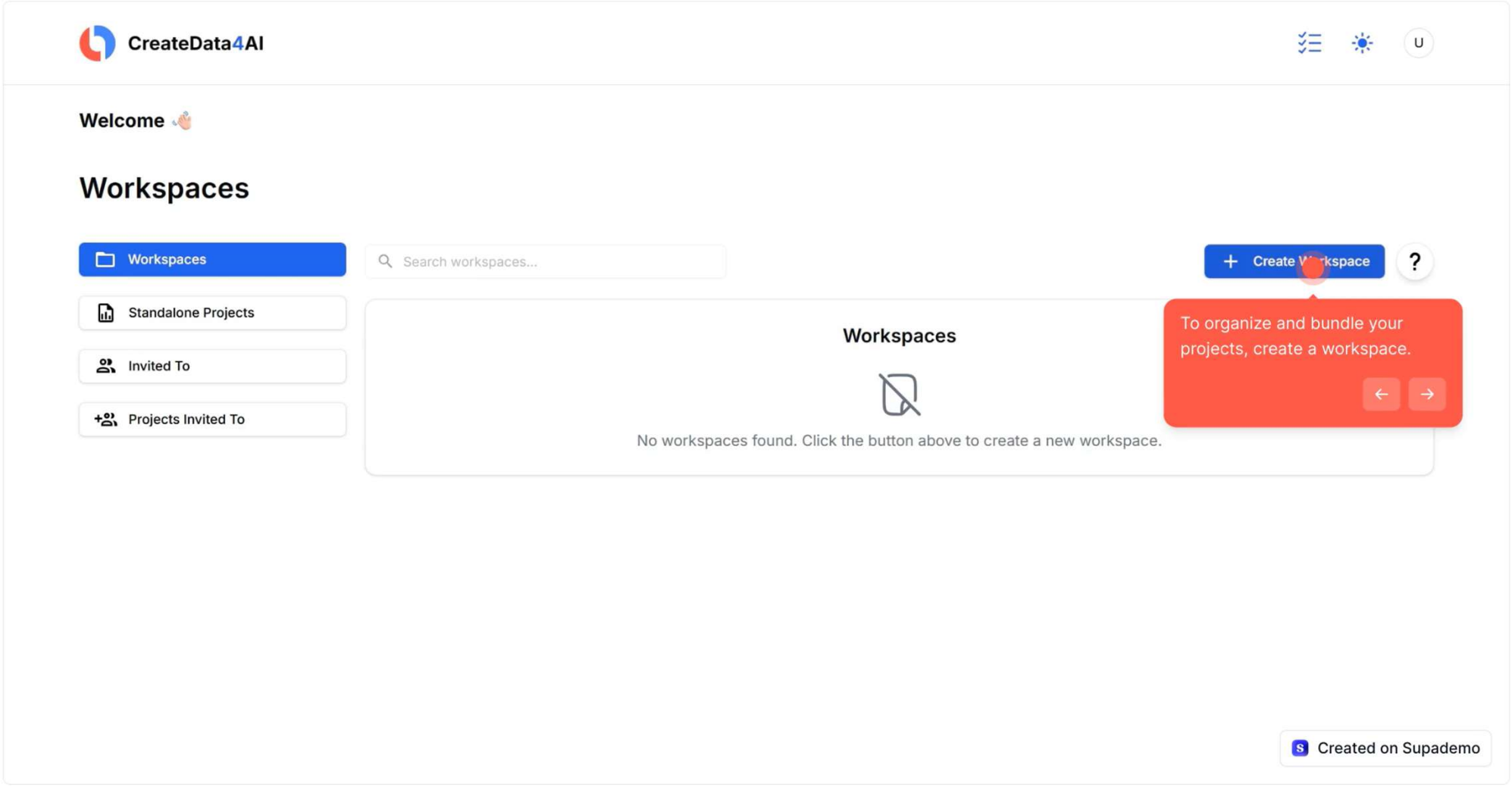
### Landing Page

Intends to improve user expectations and conceptual understanding.

### Refined Onboarding Wizard

Provides more detailed information about the conceptual specifics.

# Example: Demonstration Videos



# Results | Final Remote User Testing (Phase 4)

**In phase 4**, remote test users completed an exemplary task **without guidance**.

## 1 Recruitment via Prolific

- 15 Strangers (Prolific)
- 67% Technical Background
- Paid Participation

## 2 Test Task Completion

- NYT news article classification (400 samples, 2 classes)

## 3 Follow-up Survey

- System Usability Scale
- Feature-related questions
- Free-text questions

# Results | Final Remote User Testing (Phase 4)

**In phase 4**, remote test users completed an exemplary task **without guidance**.

**SUS Score**

72.8

↗ + 9.8

*indicates **GOOD** usability*

**Task Success Rate**

93%

↗ + 13 pp

*14 out of 15*

**Task Completion Time**

32.5 min

↗ + 3 min

*vs. 11.5 min “expert” TCT*

## Strengths

- Layout & UI
- Demonstration Videos
- Ease of Use

## Weaknesses & Issues

- Getting Started & Conceptual Understanding
- Navigation
- Context Window Assessment

# Discussion | How to Deal with Varying Usability Scores?

A significant **difference** between **technical** and **non-technical participants** can be observed.

| | Phase 2 | Phase 3 | Phase 4 |
|---|---|---|---|
| **Overall SUS Score** | 75.6 | 63.0 | **72.8** (*GOOD*) |
| **Tech. Participants** | 75.6 | 77.5 | **80.0** (*EXCELLENT*) |
| **Non-Tech. Participants** | – | 41.3 | **58.5** (*OK*) |

## Why is this?

- **Tech. users** are presumably **more familiar** with the concept of data labeling and NLP.
- **Non-tech. users** have issues with the **conceptual understanding** of the E2E process.

## How to interpret this?

- CD4AI is designed as a **collaborative platform** with different personas.
- **Tech. users** take the role of *Facilitators*[1] administer the E2E labeling workflow.
- **Non-tech. users** take the role of *Practitioners*[1] and participate in tasks only selectively.

*1) Leimeister, Jan Marco. (2014). Collaboration Engineering - IT-gestützte Zusammenarbeitsprozesse systematisch entwickeln und durchführen.*

# Discussion | What Does This Mean for the Other Research Questions?

**RQ1: What efforts can be taken to reduce manual effort in the CD4AI web application?**

- **Workflow Level:** notification system, dashboard, automated collaboration, …
- **Action Level:** smoother input operations, optional input fields
- Challenge: Overall app design is largely determined by initial design and the CD4AI framework!

**RQ3: What technical adaptations are required to transform the CD4AI […] app into an externally usable system?**

- Pipeline finalization
- Deployment, GPU acceleration, public accessibility, …

# Limitations & Future Work

## Limitations

### Methods

**Artificial Test Setup**
Authenticity of the test tasks and participants' natural motivation to achieve the given goal is limited.

**Choice of Metrics**
SUS is a widely applicable, yet generic metric. Domain-specific metrics provide more fine-granular insights.

**Participant Demographics**
Involving a larger and more diverse pool of test users and HCI experts might yield additional findings.

### Scope

**Ergonomics and Usability**
Technology adoption is also driven by usefulness, i.e., accuracy of the results and ease of system integration.

**Collaboration Features**
Testing focused on CD4AI's core user journey. Cross-team collaboration was not investigated.

## Future Work

**Feature Expansion**
Advanced dataset features, scalable infrastructure, …

**In-Depth Case Study**
Monitored long-term authentic case study with small number of test users.

**Generalization to HITL Systems**
How do results generalize to HITL systems in general?

# Conclusion

## User-centered Design

Optimized user-centered design proved a high level of usability.

## Feature Maturity

Finalized the app and enabled real-world usage.

## Public Accessibility

Made the app publicly accessible and fostered user empowerment.


B.Sc.
**Mathis Hoffmann**

Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

mathis.hoffmann@tum.de
wwwmatthes.in.tum.de

# Backup

# The System Usability Scale (SUS)

Table 3.1.: SUS survey with adapted wording for CD4AI, along the best possible score (BPS) for each item.

| ID | Question | BPS |
|---|---|---|
| Q1 | Given the need to label data, I think that I would like to use CD4AI frequently. | 5 |
| Q2 | I found CD4AI unnecessarily complex. | 1 |
| Q3 | I thought CD4AI was easy to use. | 5 |
| Q4 | I think that I would need the support of a technical person to be able to use CD4AI. | 1 |
| Q5 | I found the various functions in CD4AI were well integrated. | 5 |
| Q6 | I thought there was too much inconsistency in CD4AI. | 1 |
| Q7 | I would imagine that most people would learn to use CD4AI very quickly. | 5 |
| Q8 | I found CD4AI very cumbersome to use. | 1 |
| Q9 | I felt very confident using CD4AI. | 5 |
| Q10 | I needed to learn a lot of things before I could get going with CD4AI. | 1 |

# Inspection Phase Features

Table 4.3.: Overview of resolved defects and inconsistencies in the inspection phase.

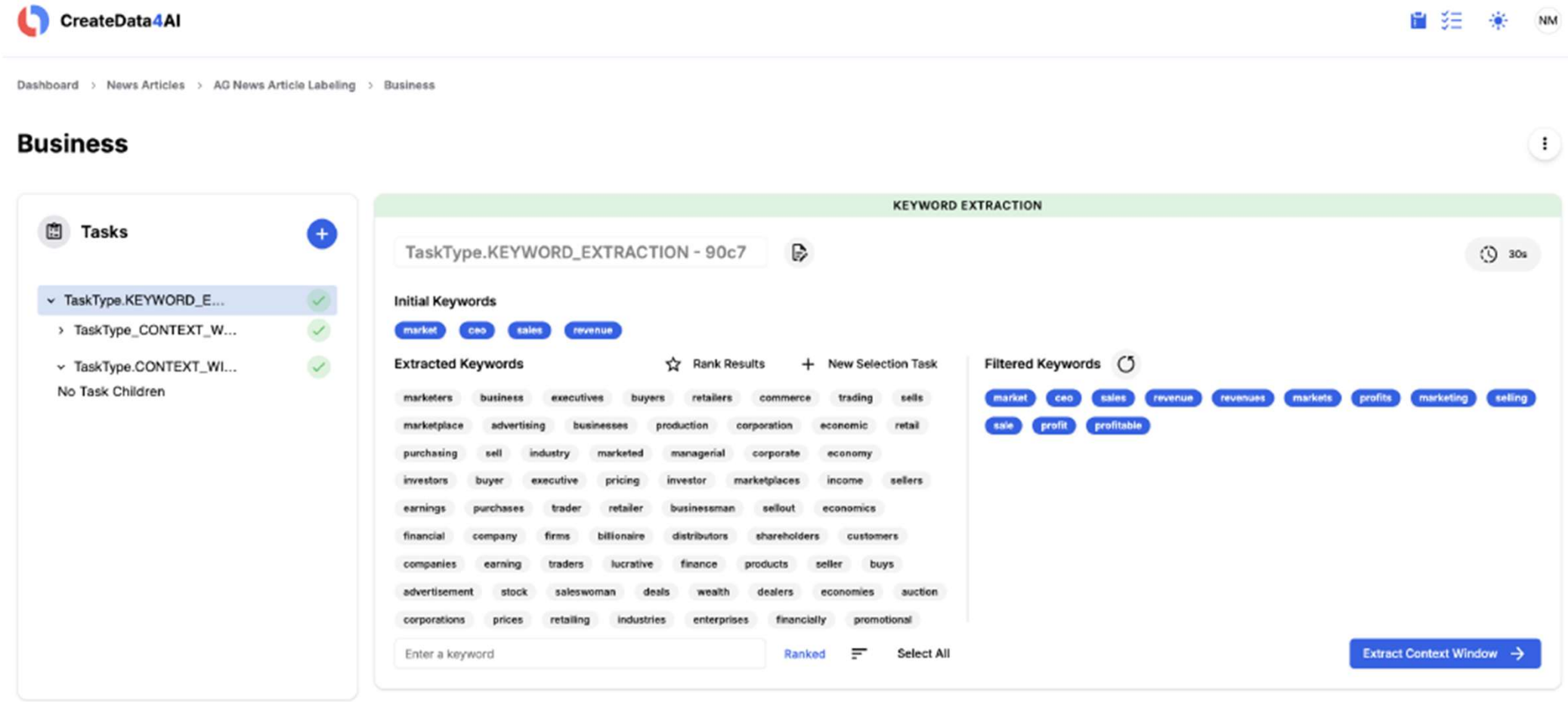
| Issue | Problem | Resolution |
|---|---|---|
| F-1.14 | Search bars existed in some views (projects, classes) but were non-functional; absent in others (workspaces, results). | Implemented fully functional, consistent search bars across workspaces, projects, classes, and results. |
| F-1.15 | The task cancellation function was not implemented; although a confirmation message was displayed, no actual cancellation occurred on the backend. | Implemented full backend integration for task cancellation, ensuring that user-initiated cancellations are now properly executed and reflected in the system state. |
| F-1.16 | The API rejected Windows-originating `.csv` files during project creation. | Ensured operating system-independent acceptance of `.csv` files. |
| F-1.17 | The UI allowed to select files with arbitrary types. File rejection only happened inside the API. | Restricted frontend input to `.csv` only. |
| F-1.18 | Mandatory form fields were not visually distinguished from optional ones. | Introduced explicit markers for required input fields. |
| F-1.19 | Auto-generated task names falsely used raw Enum keys from the source code (e.g., `TaskType.KEYWORD_EXTRACTION`). | Converted enum values into readable task labels (e.g., *Keyword Extraction*). |
| F-1.20 | The creator of a selection task was unable to reassign their own subtask, only those assigned to others. | Enabled reassignment of all subtasks, including those initially assigned to the task creator. |
| F-1.21 | Numerous inconsistencies across UI elements (naming, color and design of several buttons and pages). | Standardized styling and alignment across all views. |
| F-1.22 | Unstably implemented on-demand logic for task loading caused tasks to randomly vanish from the task tree. | Replaced on-demand loading with static logic, guaranteeing consistent task visibility. |

# Thinking Aloud Issues

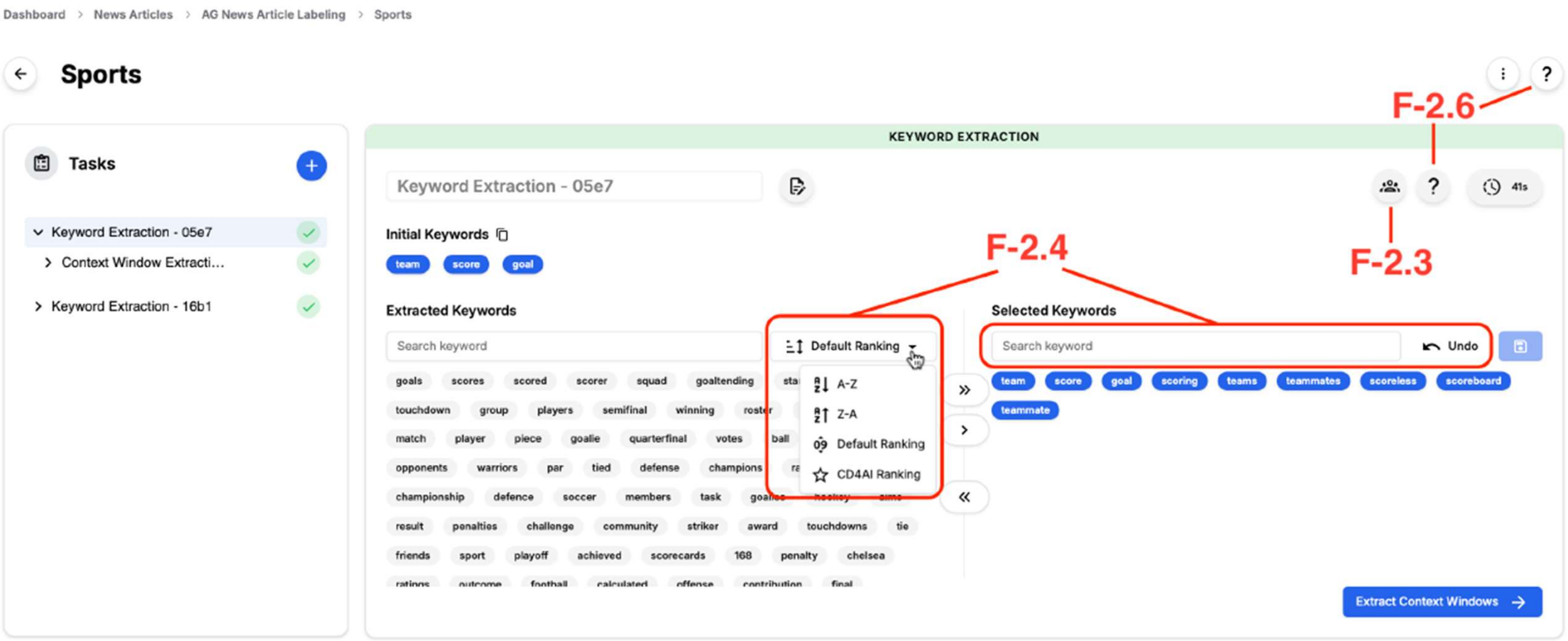
Table 4.5.: Think-aloud issues (TAIs) identified during in-person user testing.

| ID | Description | # |
|---|---|---|
| TAI 1 | No processing time shown for extrapolation tasks, reducing traceability and consistency. | 1 |
| TAI 2 | Desire for an *Undo* button to revert keyword selections. | 1 |
| TAI 3 | One email per finished task caused overload (approx. 10 per project). | 2 |
| TAI 4 | Missing status banner for the *Queued* status, and dissatisfaction with the *Queued* status icon. | 2 |
| TAI 5 | A user explicitly requested a guided tour explaining the overall workflow before the first task. | 1 |
| TAI 6 | Confusion about how to proceed when all generated archetypes looked good and no manual action was needed. | 2 |
| TAI 7 | When saving task selections, the button to start the next task appears in the same place, causing confusion. | 1 |
| TAI 8 | In light mode, the tab buttons used across the entire app have a light gray background color when unselected, making them appear like disabled functions. | 2 |
| TAI 9 | Unclear meaning and metric behind keyword *Rank* button. | 2 |
| TAI 10 | Users did not understand the dropdown to choose an embedding model, as only one option was available per language and no inline information was provided. | 2 |
| TAI 11 | The navigation from the task back to the project was confusing; one user misinterpreted the *Cancel* button. | 3 |
| TAI 12 | The tab logic on the task panel was mistaken for action buttons, leading to uncertainty about task completion flow. | 3 |
| TAI 13 | Selecting files and columns for keyword extraction was unintuitive. | 2 |
| TAI 14 | Some users did not use the batch-create option because they did not understand the button label. Others overlooked the option to add more than one task inside the batch-create dialog. | 4 |
| TAI 15 | Users did not find the function to start the extrapolation task. When starting the task, one user mentioned not understanding what they were doing there. | 5 |
| TAI 16 | One user reported a missing status toast after project creation. | 1 |
| TAI 17 | Users were frustrated that new workspaces or projects were not shown until manual refresh. | 2 |
| TAI 18 | One user's name was displayed as *None* after GitHub login. | 1 |
| TAI 19 | Users did not notice the option to add more classes in one go and created classes one by one instead. | 4 |

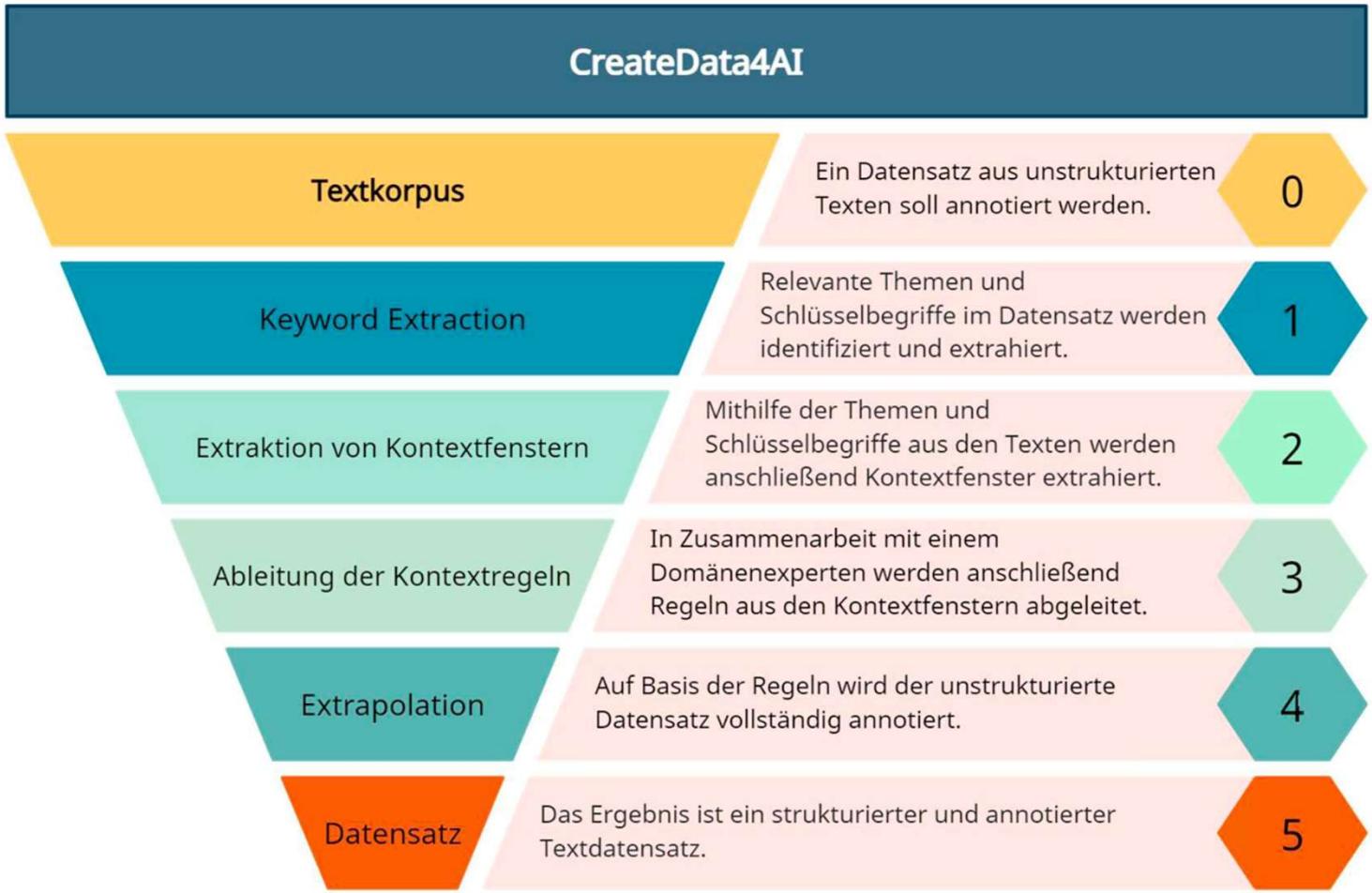
# Example: Keyword Task Panel **BEFORE**



# Example: Keyword Task Panel **AFTER**



# CD4AI – Hybrid Approach



# CD4AI – Hybrid Approach

1. **Keyword Extraction**: using classes or tags defined by a domain expert, unsupervised techniques for the extraction of keywords will be utilized to support the domain expert in defining the class. Moreover, related words and phrases are suggested to refine further the scope of the class.

2. **Context Window Extraction**: building around the keywords and keyphrases, windows encapsulating the context of these key units of information are extracted. Such windows should best embody the meaning of the selected keyword in context, so as to determine its function in text.

3. **Context Rule Creation**: using the extracted context windows, the domain expert is put into action, where he or she will determine which of the windows best describe the meaning of the predefined classes. The set of these selected rules will form the basis for automated dataset creation.

4. **Extrapolation**: for each class, the set of context rules are leveraged in conjunction with NLP techniques to "extrapolate" from the finite set of rules to a theoretically infinite number of unseen documents. With this step, the bridge between manually-defined rules and fully-unsupervised classification is crossed.

# Usability according to ISO 9241-11

- **Effectiveness**: The accuracy and completeness with which users achieve specified goals.
- **Efficiency**: The resources expended (such as time, effort, or cost) in relation to the accuracy and completeness of goals achieved.
- **Satisfaction**: The degree to which user needs and expectations are fulfilled, and the comfort and acceptability of using the product or system.

# Usability according to Nielsen (2012)

Usability is defined by **5 components**:

- **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
- **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

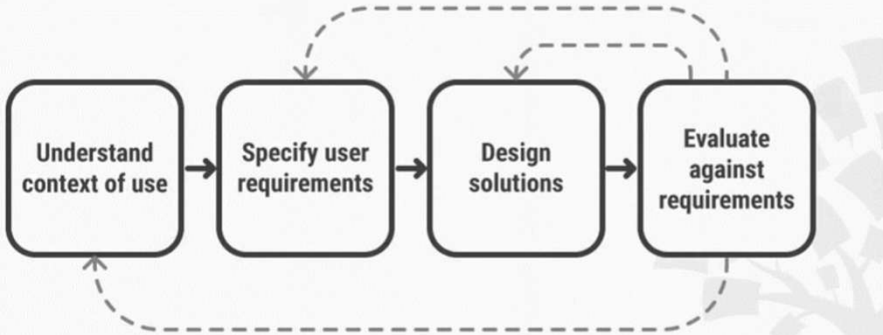
# HCI Theory – User Centered Design

## What is User Centered Design (UCD)?

User-centered design (UCD) is an iterative design process in which designers focus on the users and their needs in each phase of the design process. In UCD, design teams involve users throughout the design process via a variety of research and design techniques, to create highly usable and accessible products for them.

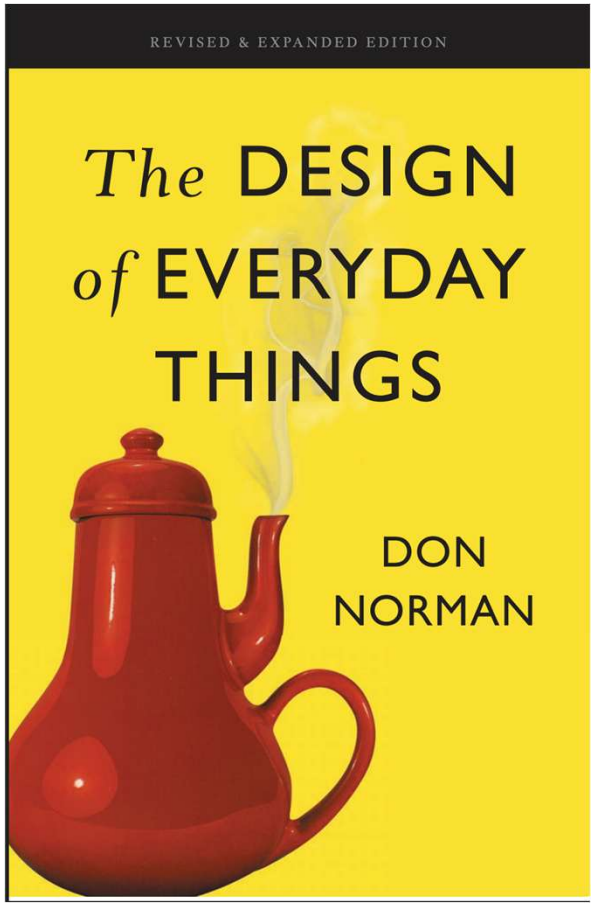
## 4 Phases in User-centered Design

Generally, each iteration of the UCD approach involves four distinct phases. First, as designers working in teams, we try to understand the *context* in which users may use a system. Then, we identify and specify the users' *requirements*. A *design* phase follows, in which the design team develops solutions. The team then proceeds to an *evaluation* phase. Here, you assess the outcomes of the evaluation against the users' context and requirements, to check how well a design is performing. More specifically, you see how close it is to a level that matches the users' specific context and satisfies all of their relevant needs. From here, your team makes further iterations of these four phases, and you continue until the evaluation results are satisfactory.




Source: https://www.interaction-design.org/literature/topics/user-centered-design?srsltid=AfmBOooBpJ2iopAtWrSb24gQslv-MEZqGM5xRa5WWIrS1e8bK6RkPVQR

# UX Design Patterns




**1. Visibility**
The more visible functions are, the more likely users will be able to know what to do next. In contrast, when functions are out of sight, it makes them more difficult to find and know how to use.

**2. Feedback**
Feedback is about sending back information about what action has been done and what has been accomplished, allowing the person to continue with the activity. Various kinds of feedback are available for interaction design-audio, tactile, verbal, and combinations of these.

**3. Constraints**
The design concept of constraining refers to determining ways of restricting the kind of user interaction that can take place at a given moment. There are various ways this can be achieved.

**4. Mapping**
This refers to the relationship between controls and their effects in the world. Nearly all artifacts need some kind of mapping between controls and effects, whether it is a flashlight, car, power plant, or cockpit. An example of a good mapping between control and effect is the up and down arrows used to represent the up and down movement of the cursor, respectively, on a computer keyboard.

**5. Consistency**
This refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks. In particular, a consistent interface is one that follows rules, such as using the same operation to select all objects. For example, a consistent operation is using the same input action to highlight any graphical object at the interface, such as always clicking the left mouse button. Inconsistent interfaces, on the other hand, allow exceptions to a rule.

**6. Affordance**
A term used to refer to an attribute of an object that allows people to know how to use it. For example, a mouse button invites pushing (in so doing acting clicking) by the way it is physically constrained in its plastic shell. At a very simple level, to afford means to give a clue (Norman, 1988). When the affordances of a physical object are perceptually obvious it is easy to know how to interact with it.

Source: https://www.designprinciplesftw.com/collections/don-normans-principles-of-design

# HCI Theory – Hick‘s Law

The time it takes to make a decision increases with the number and complexity of choices.

**Takeaways**

1. Minimize choices when response times are critical to decrease decision time.
2. Break complex tasks into smaller steps in order to decrease cognitive load.
3. Avoid overwhelming users by highlighting recommended options.
4. Use progressive onboarding to minimize cognitive load for new users.
5. Be careful not to simplify to the point of abstraction.

Source: https://lawsofux.com/hicks-law/

# HCI Theory – Others

- Cognitive Load Theory
- Fitt‘s Law
- Ecological Interface Design
- Gestalt Principles
- …

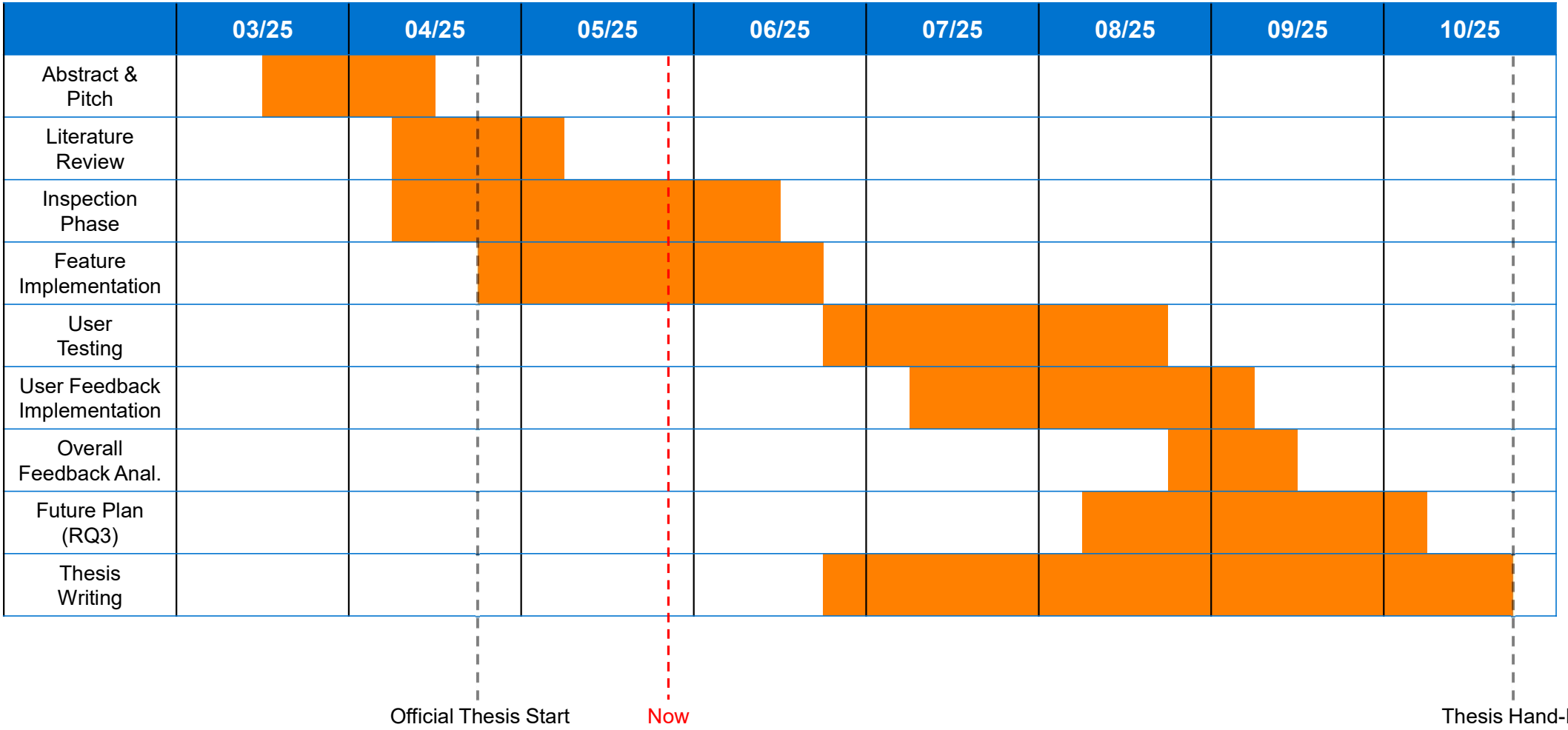
# Nielsen's 10 Usability Heuristics

Table 2.1.: Nielsen's 10 Usability Heuristics [9] with original descriptions from Nielsen [12]

| No. | Heuristic | Description |
|---|---|---|
| 1 | Visibility of System Status | "The system should always keep users informed about what is going on, through appropriate feedback within reasonable time." |
| 2 | Match Between System and the Real World | "The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms." |
| 3 | User Control and Freedom | "Users often choose system functions by mistake and need a clearly marked "emergency exit" to leave the unwanted state without an extended process." |
| 4 | Consistency and Standards | "Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions." |
| 5 | Error Prevention | "Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action." |
| 6 | Recognition Rather Than Recall | "Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed." |
| 7 | Flexibility and Efficiency of Use | "Shortcuts — hidden from novice users — may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions." |
| 8 | Aesthetic and Minimalist Design | "Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility." |
| 9 | Help Users Recognize, Diagnose, and Recover from Errors | "Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution." |
| 10 | Help and Documentation | "It is best if the system does not need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks." |

https://www.nngroup.com/articles/ten-usability-heuristics/

# Timeline from the Kickoff Presentation



# Initial Results – 3. Producing Design Solutions to meet User Requirements

Holzinger (2005)[2] has identified the three most common so-called **Inspection Methods.**
These methods are **intrinsic** without real-world users contact.

## Heuristic Evaluation

Judging to what degree the functional workflow and design elements of the app comply with established heuristics, patterns and best practices

## Cognitive Walkthrough

Task-oriented method to explore system‘s functionalities with focus on cognitive issues.

## Action Analysis

The process of breaking tasks down into actions and performing precise analytical inspection of action sequences intended to achieve a goal.

*2) Holzinger, Andreas. (2005). Usability Engineering Methods For Software Developers. Commun. ACM. 48. 71-74.*

# Feature Backlog

## Feature Prioritization

| Progress | Story |
|---|---|
| Done | Only csv files should be selectable |
| In Progress | Selection Tasks assigned to myself (the admin / assigner) are not re-assignable |
| Done | Remove validation of uncritical fields |
| In Progress | "My Tasks" are not properly loaded |
| Done | Disable Interaction Panels when Task in Progress |
| Done | CRUD Forms don't show which fields are required vs optional |
| Done | Not Found page is barely used |
| Done | Batch-select initial keywords |
| Done | Standalone projects |
| In Progress | Add option to cancel tasks |
| | Make initial keywords reusable |
| Done | Batch-create classes |
| | Data Explorer: Enable users to review a sample of the uploaded dataset |
| In Progress | Auto-start CW extraction when KW selection task is completed |
| In Progress | Auto-start extrapolation when CW selection task is completed |
| To Discuss | Task Panel: clearly differentiate between “single mode” and “collaboration mode” |
| Done | Re-design selection tasks regarding traceability, consistency |
| Done | Task progress overview page |
| Done | Email Notification System for task progress (especially reg. Archetype creation + extrap |
| | Make CW selection restartable |
| | Auto-Distribute selection task to project members (when creating KWE task) |
| | Introduce error messages |
| Must do | Add tutorial |
| Done | Class Page: add selected task to URL parameter |
| Done | Make sidebar prettier |
| Fall-off Can... | Auto-trigger CW extraction with all KW |
| Must do | Plug-in Moritz’ new algorithm |
| | Search bar is only mocked and only “available” for projects, not workspaces |
| Fall-off Can... | Advanced data exploration features; help users understand their data better |
| Fall-off Can... | Make Merge reversible |
| + New page | |

# Next Steps – 4. Evaluate against User Requirements

Holzinger (2005) also proposes the three most common so-called **Test Methods.**
These methods are **extrinsic** with contact to real-world users.

### Thinking Aloud

Test users constantly verbalize their thoughts, commenting every action they take while using the system.

### Field Observation

“Shadowing” test users during their system usage, without any kind of intervention.

### Questionnaires

Questioning test users about their opinions regarding certain aspects of the system. No direct study of the actual interface.

Holzinger, Andreas. (2005). Usability Engineering Methods For Software Developers. Commun. ACM. 48. 71-74. 10.1145/1039539.1039541.

# Stick to the good sebis traditions

- Provide action links at the bottom of the slide to guide the audience to our web pages or publications (see below). (Select the text, press CTRL-K)

- Use a file name according to our sebis conventions which helps us and our audience to find the file of your presentation on our web site with Google search:
  - YYMMDD Author Short Title
  - Include this string in the footer (Einfügen -> Kopf- und Fusszeile -> Fusszeile)
  - The unusual date format simplifies the search for the latest version of a slide in an alphabetical directory listing (Dropbox, Explorer, Tricia, Sky-Drive)

*[Ha13g] Hauder, M., Roth, S., Matthes, F.: Current Tool support for Metrics in Enterprise Architecture Management*

*For more information visit BEAMS , EAM Pattern Catalog and EAM KPI Catalog (http://wwwmatthes.in.tum.de)*

# Use the sebis visual language (shapes, fonts, colors, sizes)