MAVI KEDI

Quality Assurance

Software development project
Summer term 2024

FEN - Richard Baker

Customer: BitExpert AG

24.06.2024

Authors:
Philipp Wäsch
Zehra-Fikriye Gönenç

table of contents

1. Introduction

The software development project is a project semester for computer science students in their 4th term at Mannheim University of Applied Sciences. Groups of six students are formed to develop a software product for an external client according to requirements over the course of the term.
The task is to optimise the user interface (UI) of the prototypes created by *Cody* using *LLM*s in order to make the product more appealing to clients.

To this end, the plan is to actively involve future end users in the design process of the website. This is initially done with a questionnaire, from which the information for a prompt is taken in order to let the AI generate the first "basic page". Further specific customisations can then be made by using an AI-chatbot.
In this way, we aim to create an agile and efficient development environment that allows us to respond flexibly to the needs of the end users and deliver innovative solutions.

As part of the software development project in the summer term 2024, bitExpert AG was chosen as the client.
bitExpert AG is a company based in Mannheim with over 60 employees and is designed to develop digital solutions for different companies.

Specific must-requirements

- The website created must offer an improved user experience (UX) compared to the website generated by Cody. This includes an appealing design, intuitive user guidance and efficient functionality.
- The LLM must be fully compatible with Node.js and React.js to ensure seamless integration into the development environment.
- The chatbot should be able to edit the website live and provide direct feedback. This means that users can make changes in real time while the AI provides immediate visual feedback. This feature enables efficient collaboration between developers and designers as well as faster iteration and customisation of the web design.
- If the user has made a change using the chatbot, they can undo this using a button.

Specific nice-to-have-requirement

- The created UI prototype can be exported in different formats to be used for presentations or other development phases.

(Information 1: The words in italics are described in more detail in a glossary, which can be found on page 14)
(Information 2: The texts marked in turquoise are the adjustments based on Mr Baker's feedback: See 9.2 Schedule for permanent manual tests and 10 Glossary)

2. Project scope

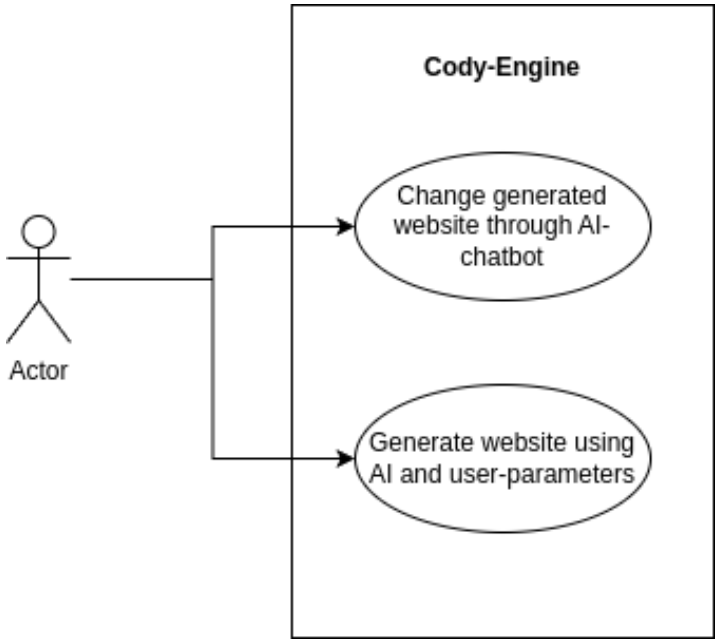

Image 1: Use-case-diagram

Use of open source LLMs

Only open source LLMs may be used for the project.
Since proprietary LLMs are chargeable, BitExpert decides against making them available.

Use of the Cody backend

Only the backend of the Cody Engine may be used for the project
In order to obtain the same functionality as the website created by the Cody Engine, the backend of the Cody Engine, which was written in TypeScript, must be used.

3. Testing strategy

Functional testing

Each function is tested.
Initially, the feature to be tested is identified. Then, the initial state, if present or relevant, is determined. After the test inputs are provided, it is verified whether the expected result has been achieved and whether the expected final state, if applicable, has occurred.

User acceptance testing

During each client meeting, the software application is presented to the client. Subsequently, the client communicates their requirements and suggestions for enhancements to us. This process ensures that the client remains informed of the continuous progress. Additionally, it serves the purpose of managing the client's expectations realistically and facilitates direct feedback on the product.

4. Test plan

Automated testing is not needed because of permanent manual testing in the development. Further errors are displayed in the frontend of the product.

ID	Test Name	Description	Success Criteria
T01	Forest-Template	Select the “Forest-Template” and click on “generate”	The “Forest-template” is used on the generated website
T02	Chatbot-response	Prompt the chatbot to change the website	The chatbot responds and the website changes
T03	Template Selection	Test if the template selection page opens when the "cody-play" button is pressed	The template selection page opens
T04	*Ollama*-API offline	Verify if an error message is displayed when the Ollama API is unreachable	An error message appears indicating API failure
T05	Undo Button	Click on the “undo”-button	The previous page is restored
T06	Redo Button	Click on the “undo”-button, then click on the “redo”-button	The previous page is restored
T07	Functionality	Test the “add book”-cody-function	A book appears in the book list
T08	Security	Enter the following JavaScript code into the chatbot: <script> alert(“cross-site-scripting detected”); </script>	No alert is displayed
T09	Colours	Select the colour “green” on the selection-site	The generated website primarily appears green

T10	Cross-Browser Compatibility	Test the application on Chrome, Firefox and Safari	Application functions correctly in all tested browsers

Table 1: Quality objectives

5. Quality metrics

Our product must fulfil the following quality criteria for bitExpert.

Quality objectives	Description
Reliability	The system should perform its functions reliably and error-free to provide a consistent and stable user experience.
Changeability/Maintainability	The system should respond to improvements, corrections, and extensions in a customisable manner.
Efficiency	The software should have a maximum loading time of 15 seconds to display the desired result and should run smoothly without errors occurring.
Compatibility	The system should work on the four most commonly used browsers, with the percentages representing the market share of the browsers (as of May 2024) Google Chrome (43.25%), Safari (42.59%), Firefox (7.08%), Edge (6.07%). It should be ensured that all functions and content are displayed correctly and run without errors. (*Source*)

Table 2: Quality objectives

6. Risk management

6.1 Risk management in our project

In order to optimise risk management, potential risks in various activities are mainly identified in the weekly sprint planning meetings and during the sprint.

These are discussed and recorded in the team. Our risk management works as follows:

- **Risk identification:** our daily Meetings provide an opportunity to address new risks and assess them.
- **Risk assessment:** Risks are prioritised based on their likelihood and potential impact. The team estimates the impact and likelihood of occurrence for the identified risks.
- **Risk planning:** For each identified risk, the team develops strategies for risk defence or risk mitigation.

6.2 Risk matrix

6.2.1 Risk matrix

The risk matrix is used to measure the extent to which the risk affects the project.

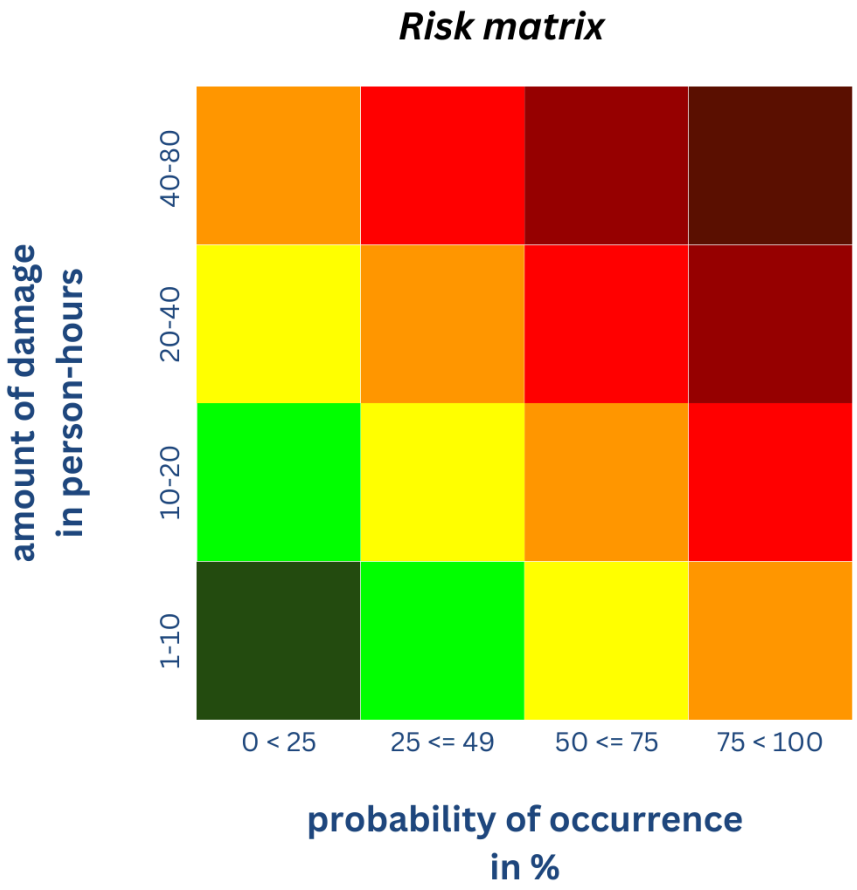

Image 2: Risk matrix

Amount of damage in person hours:

- 1-10 h: The amount of damage is minimal and hardly noticeable
- 10-20 h: The risk is noticeable due to the amount of damage, but it is not significant
- 20-40 h: The risk is decisive for the further course of the project.

- 40-80 h: The risk has a serious impact on the course of the project and has consequences for the quality of the product

Probability of occurrence in percent:

- 0 < 25: It is highly unlikely that the risk will occur.
- 25 <= 49: The risk is unlikely to occur.
- 50 <= 75: The risk is likely to occur.
- 75 < 100: It is assumed that the risk will occur.

6.2.2 Previous risks that we had

The following three risks were most significant in the project so far.

1. **Risk from 24.04.24:**
   Processing time of prompts that are too long

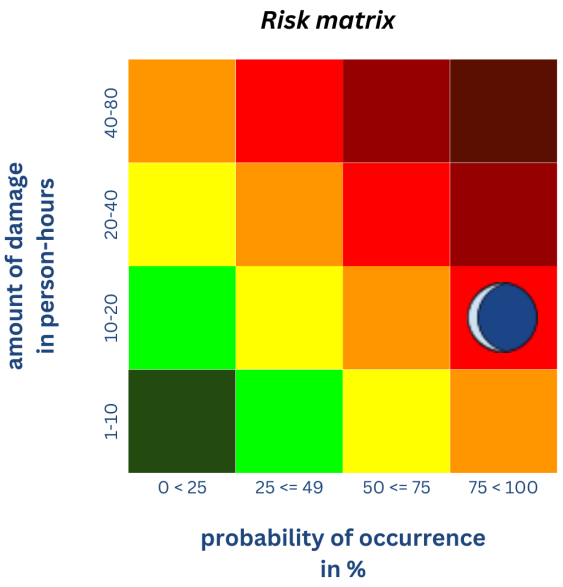

Image 3: Risk matrix from 24.04.24

2. **Risk from 08.05.24:**
   Failure of the home server (Ollama as server application)

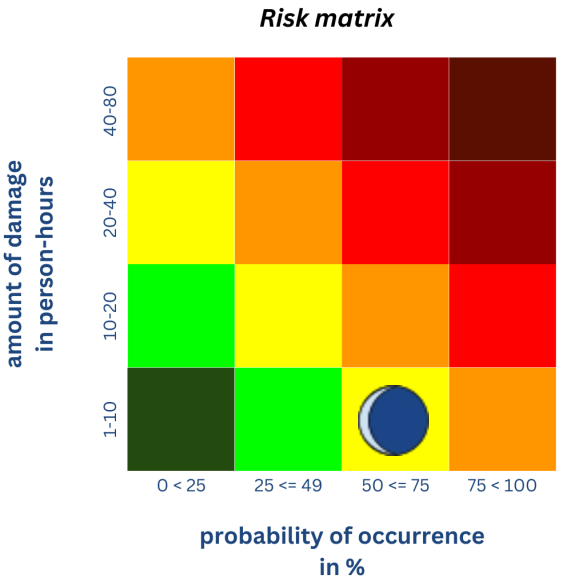

Image 4: Risk matrix from 08.05.24

3. **Risk from 29.05.24**
   AI does not provide reliable output for complex website templates

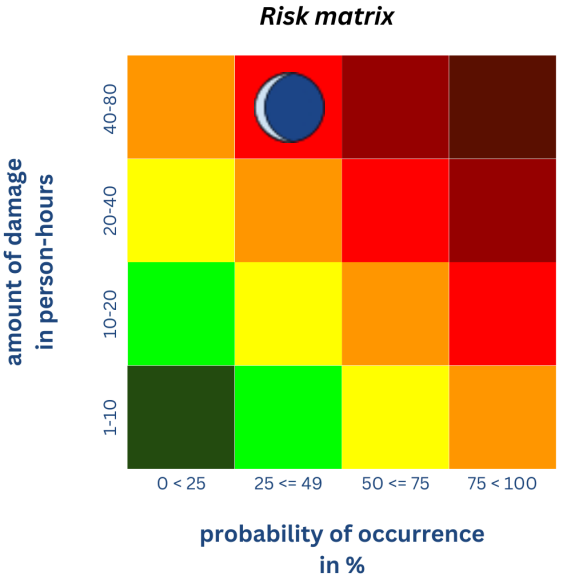

Image 5: Risk matrix from 29.05.24

7. Communication plan

Communication Plan - MAVI KEDI

COMMUNICATION GOALS	COMMUNICATION METHOD	AUDIENCE	FREQUENCY	OWNER
Daily Meetings: A daily is a short daily meeting of a team to discuss the progress of work, planned activities and possible difficulties.	**Tuesdays–Thursdays:** In present **Mondays and Fridays:** via discord	MaviKedi-Team	Every day	MaviKedi-Team
Sprint-Planning: At the sprint planning meeting, the work that needs to be done in a sprint is planned. Define Milestones.	**Tuesdays:** In present	MaviKedi-Team	Weekly	MaviKedi-Team
Sprint-Retrospective: In the retrospective, it is discussed whether the sprint was successful and try to identify potential improvements for the next sprint.	**Wednesdays:** In present	MaviKedi-Team	Weekly	MaviKedi-Team
Jour Fixe Presentation on status and Milestone updates	**Wednesdays at 10am:** In present	Professors, Tutors	Every two weeks	MaviKedi-Team
Reviews Jour Fixe and prototyp live demonstration	**Wednesdays at 10am:** In present	Professors, Tutors	Every two weeks	MaviKedi-Team
FEN-Presentation	**Friday at 9am:** In present	teaching supervisor Richard Baker	on 7 June	MaviKedi-Team
Customer meeting define new requirements, live demonstration, FAQ	**Thursdays at 10:30am:** via Microsoft Teams	Customer bitExpert	Weekly	MaviKedi-Team
Customer-final presentation	at the customer's office, **time still undefined**	Customer bitExpert	on 1 July	MaviKedi-Team

Image 6: Communication plan

8. Roles and responsibilities

8.1 Quality Assurance Responsibilities

There are two people mainly responsible for quality assurance:

- Zehra-Fikriye Gönenç
- Philipp Wäsch

Basic Principle

The document or parts of the document must only be checked and approved by a team member who was not involved in their creation.

Exceptions

Other team members will be deployed if a deadline is imminent and it is not possible for one of the above-mentioned persons to review the document in time. Another team member can review the document if they have the necessary expertise and were not involved in the creation of the document.

8.2 Documents

Code Guidelines

- Responsible: Philipp Wäsch
- Before each review, the code is checked and adjusted according to the code guidelines.

Checklists

- Responsible: Zehra-Fikriye Gönenç
- To systematically review documents and quickly provide a basis for other team members to check documents, we have created checklists. This ensures that criteria are not forgotten and that documents are consistently reviewed based on the defined criteria.

8.3 Document Control-Procedure

1. Content Review: Initially, documents are checked for content.
2. Checklist Review: Subsequently, documents are reviewed using checklists covering aspects such as spelling, grammar, formatting, and other criteria. Presentations are evaluated based on different criteria compared to other documents. Each document has its own checklist with specific review criteria.
3. Discussion of Changes: Changes are noted and discussed with the team member who created the document.

Uploading Documents

Once the responsible person has finally reviewed and approved the document, they upload it to Moodle and the MaviKedi Website.

8.4 Review Process

During each review, the product is to be presented in its current state. The team member responsible for the technical part of the review records a video of the product to show it in case of a system failure. Another team member responsible for the non-technical part of the review checks the required equipment for the presentation, such as the projector. Both presenters practise the presentation together at least once.

8.5 Usability

Usability is evaluated through questionnaires, for which Philipp Wäsch is responsible. Additionally, we test usability using techniques like the "think-aloud" method. Each team member has two independent persons, who do not have computer science knowledge, testing the product.

8.6 Code Functionality

Johannes Moseler is responsible for ensuring that the product functions as it should. Each new piece of completed code undergoes a code review in the form of a walkthrough with Colin Zenner. To avoid errors, pair programming is used when necessary.

9. Timeline

9.1 Sprint Quality Assurance Schedule

In each sprint, the product is further developed and the increment is integrated into the system. The QA schedule includes the following steps:

1. **Integration testing and system testing**
    - Each increment is initially tested for its functionality. Once verified, it is integrated into the existing system and tested again to ensure seamless integration with the current codebase. Finally, the entire system undergoes testing to confirm that all components work together harmoniously and that the overall functionality is preserved.
2. **Documentation Review**
    - Alongside technical testing, all relevant documents (see Roles and Responsibilities) are reviewed and updated.
3. **Client Review**
    - The product is presented to the client in regular meetings to ensure the project is on the right track and that changes meet the client's requirements.

9.2 Schedule for permanent manual tests

Automated testing is not needed because of permanent manual testing in the development. For the specific test that we check, see the *test plan*.

When do we conduct manual testing?

1. Management Reviews

- **When:** Tuesdays, in an interval of two weeks, before every management review

2. Client reviews

- **When:** Every Wednesday, one day before client reviews

3. *git-merge*

- We also always test manually when a team member makes changes to the code and has "merged" them.
- **When:** Whenever a modification is made

4. Final Presentation

- **When**: 24/06/2024, one week before the final presentation

9.3 Milestones and Deadlines for Testing Phases

1. Role Assignment, Tool Setup, and Team Training

- Period: 10/04/2024 - 16/04/2024
- Activities: Define team roles, set up necessary tools, and train team members on these tools.

2. Project Handbook and Requirements Specification

- Deadline: 23/04/2024
- Activities: Create the project handbook and requirements specification and complete the first version of the prototype.

3. Prototype Version 2

- Deadline: 07/05/2024

- Activities: Revise and improve the first prototype based on initial feedback rounds and internal tests.

4. Architecture Documentation Version 1

- Deadline: 14/05/2024
- Activities: Create the first version of the architecture documentation, detailing the system structure and technical specifics.

5. Template Testing

- Deadline: 28/05/2024
- Activities: Test the created templates for functionality and adaptability to various use cases.

7. Evaluate Usability Questionnaires

- Deadline: 10/06/2024
- Activities: Analyse and evaluate the collected usability data to derive concrete improvement suggestions.

8. Final System Test and Product Delivery

- Date: 28/06/2024
- Activities: Conduct the final system test to ensure the entire system functions correctly and meets the requirements. Final presentation and handover of the product to the client.

10. Glossary

Concept	Description
Cody	Cody is a bot or agent that is able to translate an event map design into working software.
git-merge	The git merge command is used to merge the changes from another branch into the current branch. This merges the development branches in order to integrate the work and achieve a standardised status. If conflicts occur, these must be resolved manually before the merge can be completed.

Git	Git is a distributed version control system that tracks versions of files. It is often used to control source code by programmers collaboratively developing software. (*source*)
LLM	A Large Language Model, or LLM for short, is a language model that is characterised by its ability to generate unspecific texts. (source)
Ollama	Ollama is an AI tool designed to help with the local execution of large language models. With Ollama, you can easily customise and create language models.
Scrum	Scrum is a process model for project and product management, especially for agile software development. (*source*)
TypeScript	TypeScript is a scripting language.

Table 3: Glossary

11. List of tables

12. List of images