

Wprowadzenie

Laboratorium #1

dr Witold Bołt, 09.10.2024 r.

Kwestie organizacyjne

- **Laboratoria**

- Spotykamy się co **2 tygodnie**, sala 3.04 - grupy mają “na zmianę”
- Zajęcia **poranne** zaczynają się o **8:30** i trwają do **10:00**
- Zajęcia **popołudniowe** zaczynają się o **12:15** i trwają do **13:45**

- **Wykład**

- Spotykamy się co **tydzień** - **nawet jeśli nie masz lab, masz wykład!**
- Wykłady trwają w godzinach 10:15 - 12:00 (*mniej więcej;*)
- Sala 1.15

Komunikacja

- **e-mail:** Witold.Bolt@ug.edu.pl
- **Github:** <https://github.com/houp/wzi2024>
- **Teams:** pojawią się niebawem grupy na Teams - info na wykładzie
 - Tu będą pojawiały się kluczowe informacje, np. o **odwołanych** zajęciach
 - Sprawdzaj grupę na Teams **zawsze** w środy rano, **przed** wyjściem z domu
 - ... inaczej możesz siedzieć przed pustą salą i czekać na *nic* :)

Zaliczenie

- Projekt grupowy w grupach **2-4 osobowych**
- Podzielony na **6 mniejszych zadań**
 - Rozwiązania umieszczamy na GitHub
 - W repozytorium powinno być README ze składem zespołu
 - Poszczególne zadania oddajemy na **kolejnych** laboratoriach
 - Wysyłamy 1 maila per zadanie z linkiem do GitHub i informacja, które zadanie zostało rozwiązane
 - Jeśli nie ma Was na laboratoriach - wyślijcie rozwiązanie mailem [**w wyjątkowych przypadkach**]

Projekt

- Wyobrażamy sobie, że chcemy zbudować **cyfrowy produkt** (ang. *digital product*) o dowolnym temacie. Najlepiej oparty o software. Może to być serwis web, aplikacja mobilna, aplikacja desktop lub jakaś forma usługi dostarczanej w sposób cyfrowy.
- Pomysł powinien być w *miarę* realny, ale sam pomysł jako taki **nie będzie oceniany** - tzn. to czy pomysł jest bardzo kreatywny, innowacyjny, realistyczny sam w sobie, wykonalny - nie będzie ocenianie.
- W ramach realizacji tych zajęć **nie** będziecie musieli zaimplementować tego produktu. Nie będzie konieczności pisania kodu, który naprawdę *zadziała*.
- Co zatem będziemy robić?!

Zadania

- Lista **12** zadań, z których jedno jest **obowiązkowe**, a z pośród pozostałych 11 należy wybrać dowolne **5**. Czyli w sumie rozwiązujemy **6 zadań**. Zadanie nr. 1 trzeba zrobić jako pierwsze. Pozostałe - kolejność **dowolna**, ale należy zachować logiczne powiązania między poszczególnymi zadaniami - bo opisują **ten sam projekt**.
- Forma rozwiązania:

- **Dokument** w języku angielskim lub polskim (format md, docx lub pdf)

LUB

- **Slajdy** prezentacji w języku angielskim lub polskim (format pptx lub pdf)
- Częścią rozwiązania powinny być obrazki, diagramy, grafy *etc.* Ale powinny być “opakowane” w stosowny dokument / slajdy z poprawnymi językowo zdaniami :)

Lista zadań 1/2

Numer	Tytuł	Czy obowiązkowe?
1	Project Vision and Scope	TAK
2	User Personas and Use Cases	NIE
3	Functional and Non-Functional Requirements	NIE
4	Wireframes and User Interface Mockups	NIE
5	Architecture (Logical and Physical)	NIE
6	Database Design	NIE

Lista zadań 2/2

Numer	Tytuł	Czy obowiązkowe?
7	Project Timeline and Schedule (Gantt Chart or Roadmap)	NIE
8	Risk Assessment and Mitigation Plan	NIE
9	Testing Plan	NIE
10	Delivery Process and Organization	NIE
11	Tooling Selection and Justification	NIE
12	Data Privacy and Security Plan	NIE

1. Project Vision and Scope

- **Description:** Define the project's main **idea**, goals, and objectives. Provide a high-level description of what the project will accomplish, the target audience, and the key problem it addresses. Define *non-goals* (what will not be covered in the project and why). Describe **why** the project will happen - why you think it makes sense to do it.
- **Acceptance Criteria:**
 - A clear vision statement is included.
 - Objectives and target audience are defined.
 - Scope limitations and project boundaries are outlined.

2. User Personas and Use Cases

- **Description:** Create profiles for primary user groups and describe how these users will interact with the product. Identify key user needs, behaviors, and goals. Answer why they will **want**, **need** or **have to** use your product. What will be the value for them? What will their key use cases?
- **Acceptance Criteria:**
 - At least two user **personas** are developed with relevant details (e.g., demographics, goals).
 - A minimum of three **use cases** are **documented** with clear steps and outcomes.

3. Functional and Non-Functional Requirements

- **Description:** List and describe the key functional and non-functional requirements. Functional requirements describe what the system should do, while non-functional requirements cover performance, usability, security, etc.
- **Acceptance Criteria:**
 - A prioritized list of at least **six** functional requirements.
 - A prioritized list of at least **three** non-functional requirements.
 - Requirements are **clear, concise, and testable**.

4. Wireframes and User Interface Mockups

- **Description:** Develop wireframes or mockups for key screens of the product, illustrating the user interface and user experience. These should convey the look and feel of the final product. Use a design / prototyping tool like Figma, Adobe XD or similar. If possible, make the mockup clickable. (Put static images in your solution + a link to clickable version if you have it.)
- **Acceptance Criteria:**
 - At least five wireframes/mockups for different screens.
 - Screens should cover primary functionalities.
 - Design reflects a coherent user flow and is consistent with the project's vision.

5. Architecture (Logical and Physical)

- **Description:** Design the system's architecture, including logical and physical diagrams. The logical diagram should illustrate software components and their interactions, while the physical diagram should show the infrastructure. Optionally, add notes on the specific technologies that will be used.
- **Acceptance Criteria:**
 - A logical diagram depicting software components and data flows.
 - A physical diagram showing deployment on infrastructure (cloud or on-prem) with specific servers / services / integrations etc.
 - Clear labeling and brief explanations for each component and each of the diagrams. Motivation of the design.

6. Database Design

- **Description:** Design a database schema that supports the project's main functionalities. Include tables, fields, and relationships between tables in case of a relational database or other relevant entities in case of non-relational databases (e.g., document structure / scheme). Choose appropriate format - either diagrams, tabular description, pseudo-code, example data points etc.
- **Acceptance Criteria:**
 - A database schema with key tables identified.
 - Relationships (e.g., one-to-many) are clearly defined.
 - Field names, data types, and any constraints are included.

7. Project Timeline and Schedule (Gantt Chart or Roadmap)

- **Description:** Present project team composition with key roles and responsibilities. Develop a project timeline, including major milestones, deliverables, and deadlines. This can be presented as a Gantt chart or a roadmap.
- **Acceptance Criteria:**
 - A Gantt chart or roadmap covering at major milestones.
 - Each milestone has a description, estimated time, and dependencies.
 - Clear division of work phases (e.g., planning, design, development, testing).

8. Risk Assessment and Mitigation Plan

- **Description:** Identify potential risks that could impact the project and develop a mitigation plan for each risk. Risks could include technical challenges, timeline delays, or resource limitations. For each of the risk assign the **likelihood** (LOW, MEDIUM, HIGH, VERY HIGH) - how **probable** given risk is and **impact** (LOW, MEDIUM, HIGH, VERY HIGH) - how “**bad**” for the project will it be **if** it happens.
- **Acceptance Criteria:**
 - At least five risks are identified and categorized by likelihood and impact.
 - Mitigation strategies are provided for each risk.
 - A risk matrix or table is included to summarize findings.

9. Testing Plan

- **Description:** Outline a testing strategy for the project, covering different types of tests (e.g., unit tests, integration tests, user acceptance tests). Describe how the product's functionality and reliability will be validated. Prepare few test cases with detailed description and relevant testing strategy. If possible prepare a testing schedule - when these test will happen, who will test and how long we expect the tests to take.
- **Acceptance Criteria:**
 - A list of at least three test types with brief descriptions.
 - Specific test cases for key features, including expected outcomes.
 - Description of testing tools or frameworks to be used (if applicable).

10. Delivery Process and Organization

- **Description:** Define the project's delivery methodology (e.g., Agile, Scrum, Kanban, or Waterfall) and explain how the chosen approach aligns with the project goals. Describe how the team will structure their workflow, manage tasks, and handle changes. If possible name specific tools that will be used for this.
- **Acceptance Criteria:**
 - A description of the selected methodology with key principles.
 - Rationale for choosing this approach for the specific project.
 - Outline of project phases or sprints (if Agile-based), with roles and responsibilities.

11. Tooling Selection and Justification

- **Description:** Identify the key tools and technologies used for the project, including: programming languages, frameworks, IDEs, project management software, version control systems, CI/CD systems, design tools, collaboration platforms and any other relevant tools. Provide reasoning (**why?**) for each tool's selection.
- **Acceptance Criteria:**
 - A list of tools with a brief description of each tool's purpose.
 - Justification for selecting each tool based on project requirements.
 - Description of how the tools integrate or complement each other.

12. Data Privacy and Security Plan

- **Description:** Identify potential privacy and security issues related to the product. Document how data will be protected and compliance with relevant law regulations (e.g., GDPR). Check if you need to obtain any form of certification or do any audits. Plan security related testing (e.g., penetration testing). List tools and approaches for increasing the security (encryption, network protection, etc.).
- **Acceptance Criteria:**
 - A list of at least three potential security risks is identified.
 - Mitigation strategies for each risk are outlined.
 - An overview of how data privacy / law compliance will be achieved.

Q&A

Pytania

- Czy można zmienić temat projektu w trakcie przedmiotu? **Nie.** (Ewentualnie jeśli bardzo trzeba, to można - ale trzeba wstecznie poprawić oddane już zadania.)
- Czy można zmienić skład grupy w trakcie przedmiotu? **Nie.**
- Czy można zrobić więcej zadań niż 6? **Tak.**
- Czy można zrobić mniej zadań niż 6? **Tak**, ale ocena będzie **niższa**.
- Czy można korzystać z ChatGPT i podobnych narzędzi? **Tak**, ale trzeba **rozumieć** swoje rozwiązanie i umieć odpowiadać na pytania.
- Czy można użyć projekt z innego przedmiotu? **Tak.**
- Czy można rozwiązać dodatkowe zadania - inne niż te podane na liście? **Tak**, ale po **konsultacji** z prowadzącym.

Dziękuję!

Witold.Bolt@ug.edu.pl



Współczesne
Zastosowania
Informatyki

