Exercise 4 – Reviews & Static Code Analysis Tools

Goal:

The goal of this exercise is to practice a review situation using the document of the *FocusFlow* application created so far. The document will be reviewed by a group of experts (here: you) and, if present, defects are identified and documented as a result of the review.

Exercise 4.1 (12 Points): Conduct a Review

In this exercise, you should practice the presented "review process" (planning, kick-off, preparation, meeting, …).

Your team performs a review of the following documents:

- FocusFlow introductory text (see appendix)
- the functional system requirements and specification: https://github.com/dgrewe-hse/focusflow/blob/dev/docs/spec/spec.md
- existing code base of the core entities so far: https://github.com/dgrewe-hse/focusflow/tree/dev/backend/src/main/java/de/hse/focusflow

Please select the appropriate review roles in your team (e.g., Moderator, Experts, Note-Taker, etc.), select the appropriate review type (e.g., walkthrough, inspection, technical review, …) as well as use the provided review template in the Moodle submission. Document the type of review you selected within your repository.

Please find a list of templates in the master repository: https://github.com/dgrewe-hse/focusflow/tree/dev/docs/templates/review

Exercise 4.2 (5 Points): Retrospective

Reflect on the review. What did you notice? Are reviews a suitable method for you and your team? If so, for which area would you use them?

Exercise 4.3 (10 Points): Static Code Analysis Tools for FocusFlow

Review the following list of static code analysis tool categories (e.g., use the list from the slides as well) and select two types of tools that you believe are most suitable for the FocusFlow application. Dependent on the pro
Integrate these tools into your project, document the steps, and evaluate their impact on the project directly within your repository. You are not limited to the tools listed below. Feel free to propose/integrate other tools you think are valuable in the context of the project.

List of Categories:

- Linters
  - Java: Checkstyle, PMD
  - Python: Pylint, Flake8
  - JavaScript/Typescript: ESLint
- Type checkers
  - Python: MyPy
- Security scanners
  - Java: SpotBugs
  - Python: Bandit
  - JavaScript/Typescript: npm audit
- Code coverage tools
  - Java: JaCoCo
  - Python: coverage
  - JavaScript/Typescript: Jest
- Complexity analyzers
  - Java: PMD
  - Python: Radon
  - Javascript/Typescript: ESLint (with complexity rules)
- Dependency checkers
  - Java: OWASP Dependency-Check
  - Python: safety
  - Javascript/Typescript: npm audit
- Style checkers
  - Java: Checkstyle
  - Python: black
  - JavaScript/TypeScript: Prettier
- Dead code detectors
  - Java: PMD
  - Python: vulture
  - JavaScript/Typescript: ESLint (with no-unused-vars rule)

Instructions:

- Select two categories (and two different tools) from the list above that you think would be most supportive for your implementation of the FocusFlow application. Justify your selection.
- Integrate the selected tools into the project. Document the steps you took to set up and configure the tools.
- Evaluate the impact of these tools on the project. Do you think they are useful for improving code quality and catching issues early, or do they potentially slow down the development process? Provide your reasoning.

Appendix

Introductory Text:

The FocusFlow application is designed to offer a structured and accessible solution for managing everyday tasks and enhancing personal as well as small-team productivity. It focuses on supporting individuals and small working groups in organizing their daily responsibilities, coordinating assignments, and maintaining a clear overview of work progress. By providing a user-friendly and efficient digital environment, FocusFlow ensures that users can concentrate on completing their tasks without being overwhelmed by complexity or unnecessary features.

A core element of FocusFlow is the management of individual tasks. Users are able to record their work items within the system and organize them in a way that reflects their priorities. Tasks may include titles, relevant details, and contextual information to clarify their purpose. The system supports the grouping of tasks into thematic collections or categories, which allows users to structure their workload according to specific projects or areas of responsibility. Each task is associated with a status that reflects its current state in the workflow, helping users to track progress over time.

In addition to managing personal task lists, FocusFlow is also suited to collaborative work within small teams. Users may assign tasks to one another in order to coordinate responsibilities more effectively. This allows for basic team management and a shared understanding of ongoing work without requiring complex project management tools. When tasks are assigned, users can be notified automatically, ensuring that no updates are missed. This basic interaction model supports the daily routines of teams where responsibilities frequently shift and transparent communication is essential.

Another essential component of FocusFlow is the representation of task timelines and due dates. Tasks may be linked to specific dates, enabling users to organize their work according to deadlines or scheduled events. The system is expected to assist users in identifying overdue tasks, pending items, and upcoming priorities. This supports time-awareness and contributes to the overall goal of reducing task-related stress and disorganization. While reminders or alerts may be used to draw attention to time-sensitive items, the system should avoid excessive notifications that could distract rather than support the user.

From the outset, FocusFlow is intended to remain a lightweight and focused application. Unlike many other systems which offer a wide array of features that often go unused, the goal of FocusFlow is to concentrate on essential functionality and deliver it in a clear and reliable manner. Its interface is expected to be intuitive and accessible, allowing users to begin working with the system without extensive training or prior knowledge. Whether used individually or in a small group setting, the application is meant to support rather than complicate existing workflows.

FocusFlow is also designed with accessibility in mind, as it will be available as a web-based solution. This ensures that users can access their task lists and updates from any location and on a range of devices. The system will be built using a modular structure that separates data storage, application logic, and user interface. This allows for flexible development and potential future enhancements, while maintaining a clear and maintainable architecture.

While FocusFlow is not a project planning tool in the traditional sense, it fills an important role in daily work organization, offering users the ability to capture, structure, and follow up on their commitments. The application is suitable for various user groups, including students, professionals, freelancers, and small teams, all of whom share the need for a straightforward and dependable productivity aid.

By focusing on clarity, relevance, and ease of use, FocusFlow addresses a common problem faced in modern digital work environments: the lack of practical and non-intrusive tools for managing everyday tasks. Its anticipated functionality supports the typical needs of users without overwhelming them, making it an attractive solution for individuals and teams seeking structure, clarity, and calm in their daily routines.

Summarized, the following high-level expectations should guide the requirements engineering process:

1. Simplicity and Clarity

The application must offer a user interface that is clean, logical, and easy to navigate. Users should be able to focus on their actual work, not on figuring out how the tool operates.

2. Support for Daily Task Management

There should be a practical way for users to create tasks, review them, and manage their completion over time. Tasks may vary in complexity, and users should be able to organize them accordingly.

3. Basic Collaboration Features

Although **FocusFlow** is primarily designed for individual use, it must also be suitable for small teams. Users should be able to assign or share tasks with others and maintain transparency in ongoing activities.

4. Awareness of Time and Progress

Users benefit from knowing what is due soon, what is in progress, and what has already been completed. Ideally, the system would reflect these states clearly, possibly with some form of visual feedback.

5. Minimal Disruption

Notifications and reminders should support rather than distract. The system should assist users in staying informed, without overwhelming them with excessive alerts.