American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Project Title:

Integrated Application Usage Tracking & Parental Control System

Supervised by:

TONNY SHEKHA KAR

A Software Engineering Project Submitted
By

Semester: Summer_23_24		**Section: H**	**Group Number: 09**	
SN	Student Name	Student ID	Contribution (CO3)	Individual Marks
1	A. F. M. RAFIUL HASSAN	22-47048-1	50%	
2	MD. ASHIKUZZAMAN ABIR	22-47006-1	50%	

The project will be Evaluated for the following Course Outcomes

CO3: *Select* appropriate software engineering models, project management roles and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects	Total Marks	
Appropriate Process Model Selection and Argumentation with Evidence	[5 Marks]	
Evidence of Argumentation regarding process model selection	[5Marks]	
Analysis the impact of societal, health, safety, legal and cultural issues	[5Marks]	
Submission, Defense, Completeness, Spelling, grammar and Organization of the Project report	[5Marks]	

Description of Student's Contribution in the Project work

Student Name: MD. ASHIKUZZAMAN ABIR
Student ID: 22-47006-1
Contribution in Percentage (%): 50
Contribution in the Project:

- Model Selection
- Role and Responsibility
- Impact on Society, Environment and Culture

____
Signature of the Student

Student Name: A. F. M. RAFIUL HASSAN
Student ID: 22-47048-1
Contribution in Percentage (%): 50
Contribution in the Project:

- Model Selection
- Role and Responsibility
- Impact on Society, Environment and Culture

____
Signature of the Student

1. SOFTWARE DEVELOPMENT LIFE CYCLE

1.1 Process Model

Selected Process Model and Argumentation with Evidence: The process model "Incremental Development" has been chosen for our project, "Integrated Application Usage Tracking & Parental Control System." This is also known as iterative development.

The requirements of our project are very well known. In this process model, the whole project is divided into several increments and the core product has priority. The primary product takes precedence in this process model, which divides the entire project into multiple increments. It may be possible to deliver the highest priority requirements earlier by using the incremental development process framework. The reason this process model was chosen is that there is less chance of a project failing as a whole, and the system services with the highest importance usually experience the most testing.

Since our project requirements are well known, so there is no need to update them. This is why we avoid the Agile process model. Examples include the XP process model, DSDM, and others. This is not a really big project. As a result, FDD is not chosen because it is typically used for larger projects with many developers. SCRUM is also appropriate for embedded system-related projects, so we avoid using this process model. Other SE models, such as the Waterfall process model and the V model, are not used because the Waterfall is an inflexible and linear process with a high failure rate.

1.2 Project Role Identification and Responsibilities

By using the incremental model in software engineering, a project is usually divided into smaller, easier-to-manage increments, or iterations. Every iteration improves and enhances the software incrementally while sticking to an equivalent development procedure. These are some basic roles and responsibilities within an incremental model process; however, they might change based on the particular project and team structure. In this process model, the following roles are mostly present.

1. Project Manager:

- Overall responsibility for planning, organizing, and coordinating the incremental development process.
- Define project objectives, scope, and milestones for each iteration.
- Allocate resources, manage budgets, and track progress.
- Ensure effective communication and collaboration among team members.

2. **Product Owner:**

- Represents the stakeholders and end-users.
- Defines and prioritizes features and requirements for each iteration based on business value and user feedback.
- Collaborates with the development team to refine and clarify requirements as needed.
- Accepts or rejects completed increments based on predefined acceptance criteria.

3. **Development Team:**

- Software Developers: Responsible for designing, coding, and testing the software increments.
- Quality Assurance (QA) Engineers: Ensure the quality and reliability of each increment through testing and validation.
- User Experience (UX) Designers: Design user interfaces and ensure a positive user experience.
- Technical Writers: Create documentation and user manuals for the software increments.
- DevOps Engineers: Manage the deployment, configuration, and maintenance of the software increments.

4. **Stakeholders:**

- Provide feedback on completed increments and participate in reviews and demonstrations.
- Collaborate with the product owner to prioritize features and requirements for future iterations.
- Support the development team by providing domain knowledge and resources as needed.

Overall, the incremental model encourages a collaborative and iterative approach to software development, with roles and responsibilities distributed among team members to ensure the successful delivery of incremental value to stakeholders.

Impact of the Developed Software on Society and Environment: Our developed software “Integrated Application Usage Tracking & Parental Control System” can have both positive and negative impacts on society and the environment. Here's an evaluation of its sustainability:

1. **Society:**

- Positive Impact:

  1. Increased awareness: Monitoring applications can help users become more aware of their digital habits and encourage them to limit their usage, leading to improved mental health and well-being.

2. Productivity improvement: By monitoring application usage, individuals and organizations can identify time-wasting activities and focus on more productive tasks, potentially leading to higher efficiency and effectiveness.

- Negative Impact:

1. Privacy concerns: Constant monitoring of application usage may raise privacy concerns among users, as it involves tracking their digital activities and potentially collecting sensitive data.
2. Dependency on technology: Over-reliance on monitoring tools to regulate usage may prevent individuals from developing self-discipline and managing their screen time effectively without external intervention.

2. Environment:

- Positive Impact:

1. Energy conservation: Limiting application usage can reduce the energy consumption of devices, leading to lower carbon emissions and a smaller environmental footprint.
2. E-waste reduction: Decreasing the frequency of device usage can prolong their lifespan, reducing the need for frequent upgrades and minimizing electronic waste generation.

- Negative Impact:

1. Increased server load: Constant monitoring of applications and their usage data may require additional server resources, leading to higher energy consumption in data centers and contributing to environmental degradation.
2. Electronic device disposal: Limiting application usage may lead to a longer lifespan for electronic devices, but eventually, they will still need to be disposed of, contributing to e-waste accumulation unless proper recycling measures are taken.

3. Cultural:

- Positive Impact:

1. Cultivation of digital wellness: Emphasizing the importance of balanced technology usage can foster a cultural shift towards prioritizing mental health and well-being over constant connectivity and digital consumption.
2. Promotion of cultural values: Application monitoring and usage limits can align with cultural values emphasizing moderation, self-discipline, and mindful consumption, contributing to a healthier societal ethos.

- Negative Impact:

  1. Resistance to change: Cultural norms and habits around technology usage may resist efforts to implement monitoring and restriction measures, leading to pushback and potential cultural conflicts.

  2. Technological dependency: Over-reliance on monitoring tools to regulate behavior may exacerbate cultural tendencies towards technological dependency, undermining efforts to cultivate self-control and autonomy.

In summary, the sustainability of integrated application usage tracking & parental control system requires careful consideration of its societal, environmental, and cultural impacts. Balancing the benefits of promoting digital wellness and environmental conservation with potential drawbacks such as privacy concerns and cultural resistance is essential for fostering a sustainable relationship with technology.