Automated Code Review System
(Feasibility Study)

SOEN 6841 – Software Project Management
By
Prof. Joumana Dargham

Submitted By – Group 18

NAME	STUDENT ID NUMBER
Dhruv Panchal	40226430
Udisha Kaura	40266183
Khushal Nirmal Jain	40233877
Aman Kumar	40278443

TABLE OF CONTENTS

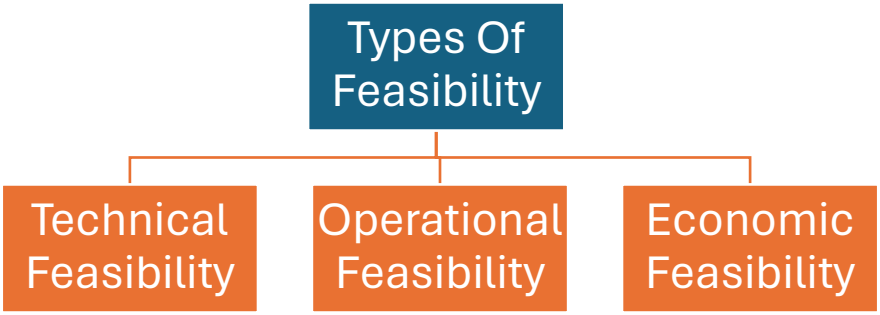
Feasibility Study

Objective

The purpose of this document is to conduct a thorough feasibility study for the **Automated Code Review System (ACRS)**, evaluating its objectives, requirements, and technical, economic, and operational viability.

Main Functionalities:

The Automated Code Review System (ACRS) represents a pivotal endeavor in modern software development, which aims to streamline code review processes and foster collaboration among development teams. As organizations increasingly rely on efficient code management practices to ensure software quality and reliability, ACRS emerges as a comprehensive solution designed to meet these evolving needs. Hereby, we outline the main functionalities of our proposed project.



- **Task Management and Version Control:** This feature addresses the need for organized planning and task management within the code review process, enhancing efficiency and productivity.

- **Document Sharing:** Enabling document sharing facilitates collaborative review and feedback, fostering communication and collaboration among team members. This functionality promotes effective team communication and collaboration, essential for successful project execution, Allowing users to set milestones and track project progress ensures project milestones are met and provides visibility into the project's status.

- **Integration with Other Tools:** Seamless integration with other productivity tools enhances workflow efficiency and productivity, streamlining the development process. The outlined development phases follow a structured approach to software development, ensuring systematic progress and adherence to project requirements:

- **Requirements Elicitation:** Gathering and documenting requirements ensures that the software meets the needs of stakeholders and users. Additionally, designing the system architecture based on requirements ensures that the software is well-structured and scalable.

- **Code Analysis and Quality Checks:** Implementing automated code analysis tools and quality checks helps in identifying potential issues such as code smells, performance bottlenecks, security vulnerabilities, and adherence to coding standards. This ensures that the codebase is of high quality and follows best practices.

1. Technical Feasibility:

Determining technical feasibility is crucial in managing our **Automated Code Review System (ACRS)**. It helps ensure that our project is doable from a technical standpoint. Checking things like the technological requirements, the availability of the technical infrastructure, and the possibility of leveraging essential tools helps us determine the viability of the proposed project. For ACRS, this means making sure the technology we are using is all in line with what we want to achieve. We consider multiple technical aspects to determine the technical feasibility as follows :

- **Development Approach:** ACRS will be created using the **Django framework**, which is a popular choice for web development. This framework ensures that the software will work smoothly on various operating systems, making it accessible to a wide range of users.

- **System Lifespan:** ACRS is designed to remain relevant and functional for approximately **5 years**. This estimation is based on the current technology landscape and anticipated advancements in the field. It provides a reasonable timeframe for the organization to benefit from the investment in developing the software.

- **Hosting:** ACRS will be hosted on **Amazon Web Services (AWS),** a leading cloud computing platform known for its scalability, reliability, and security. By leveraging AWS, the organization can easily scale the infrastructure to accommodate increasing user demand and ensure high availability of the software.

- **Resources Needed:** Building ACRS will require a skilled team of professionals, including two back-end developers proficient in Django, and three web hosting experts experienced in managing **cloud-based infrastructure**. Additionally, investing in new hardware for developers will provide them with the necessary tools and equipment to efficiently develop the software.

- **Compliance Requirements:** Ensure that ACRS complies with relevant industry standards, regulatory requirements, and data protection laws such as GDPR, HIPAA, or PCI DSS.

This includes handling sensitive data appropriately, maintaining audit trails, and providing necessary documentation for compliance audits.

- **System Lifespan Consideration:** Plan for regular updates and maintenance cycles to keep ACRS aligned with evolving technology trends, security patches, and industry best practices. Design the system architecture with modularity and extensibility in mind, allowing for easy integration of new features and functionalities over time. Conduct periodic technology assessments and feasibility studies to evaluate the need for upgrades or migrations to newer frameworks or technologies.

2. Operational Feasibility:

Ensuring the smooth operation and successful outcome of our project is paramount, mirroring the significance we place on the effectiveness of our Automated Code Review System (ACRS). This entails not only the creation of functional software but also the orchestration of processes that enable its realization in a timely and efficient manner. By prioritizing operational feasibility, we lay the groundwork for a project environment characterized by cohesion, adaptability, and a clear path toward achieving our desired outcomes.

- **Resource Availability:** Ensuring adequate availability of resources, including skilled personnel and necessary tools, is crucial for the operational feasibility of ACRS. The project will require a team of skilled professionals, including developers proficient in **Django framework** for both **front-end and back-end development**, web hosting experts experienced in managing cloud-based infrastructure, and quality assurance testers proficient in software testing methodologies. Additionally, tools such as **integrated development environments (IDEs),** version control systems, project management software, and communication platforms will be essential for facilitating collaboration and productivity. By assessing resource requirements and allocating them effectively, the project team can mitigate potential bottlenecks and ensure smooth progress throughout the development lifecycle.

- **Training and Support:** Ensuring that every member of our team is well-equipped to utilize ACRS effectively is crucial for the success of our project. We will conduct comprehensive training sessions to familiarize users with the system's features and functionality. Additionally, we will establish a robust support system, including detailed user guides, **frequently asked questions (FAQs),** and a dedicated helpdesk, to provide assistance whenever needed. By investing in thorough training and ongoing support, we can empower our team to leverage ACRS to its fullest potential, enhancing productivity and ensuring the smooth operation of our project.

- **Timely Completion:** To ensure timely completion of the project, production capacities per day per phase have been determined. This involves breaking down the project into

manageable tasks and establishing realistic timelines for their completion. By closely monitoring progress and addressing any issues promptly, the project team can stay on track to achieve project milestones within the desired timeframe.

- **Incremental Development:** ACRS will be developed using an incremental development methodology, which involves delivering the software in small, incremental releases. This approach allows stakeholders to see tangible progress at regular intervals and provide feedback that can be incorporated into subsequent iterations. By **iterative** refining and improving the software, ACRS can better meet the evolving needs of its users and stakeholders.

3. Economic Feasibility:

Making sure our project makes sense financially is key, just like with our Automated Code Review System (ACRS). In software projects, where resources are limited, understanding the economics is crucial. Careful economic documentation supports the prospects for funds allocation. Additionally, it also makes sure we make smart decisions about where to spend our money throughout the project. This documentation acts as a guide, helping us make informed choices and manage our finances wisely. Ultimately, it helps us make sure our project stays on track and achieves its goals without overspending.

- **Budget:** The organization has allocated an initial budget of **$250,000 CAD** for the development of ACRS. This budget provides a financial framework for the project and ensures that sufficient resources are available to support its implementation.

- **Total Costs:** The estimated total cost for outsourcing development tasks and salaries for the project team is approximately **$163,000 CAD.** However, in a **worst-case scenario**, where unforeseen expenses arise or project delays occur, the total cost could reach up to **$244,000 CAD**. Despite this potential variability in costs, the expected budget is anticipated to fall in this range.

- **Revenue and Profit Expectations:** ACRS is expected to generate a positive profit margin, indicating that the benefits derived from its implementation outweigh the costs incurred during development and maintenance.The revenue generation potential of ACRS stems from its ability to streamline code review processes, improve code quality, reduce development cycle times, and enhance team collaboration, leading to increased productivity and software reliability. By offering ACRS as a valuable solution to organizations facing code management challenges, there is a revenue opportunity through subscription-based models, licensing fees, or service contracts.

- **Risk Mitigation and Cost Management:** Implement risk management strategies to mitigate potential cost overruns, such as conducting thorough requirements analysis, adopting agile project management practices, monitoring project progress regularly, and having contingency plans in place. Utilize cost management techniques such as cost-

benefit analysis, resource optimization, vendor negotiations, and budget tracking tools to ensure efficient utilization of allocated funds and minimize wastage.

- **Return on Investment (ROI) Analysis:** Conduct a detailed ROI analysis to evaluate the financial benefits of implementing ACRS compared to the initial investment. Consider factors such as cost savings from reduced manual code review efforts, increased developer productivity, decreased time-to-market for software releases, and potential revenue from product sales or service offerings. Calculate key financial metrics such as payback period, net present value (NPV), internal rate of return (IRR), and profitability index (PI) to assess the project's financial viability and attractiveness.

- **Profit:** ACRS is expected to generate a positive profit margin, indicating that the potential benefits outweigh the costs associated with its development and implementation. By providing a valuable solution to a real-world problem or opportunity, ACRS has the potential to generate revenue and contribute to the organization's long-term success.

- **Conclusion:** Based on the findings of the feasibility study, ACRS is deemed technically, operationally, and economically feasible, making it a viable project for the organization to pursue. By addressing key considerations such as technology requirements, resource availability, and potential return on investment, the organization can confidently move forward with the development and implementation of ACRS.