Neighborhood Watch Incident System SDLC Model and 16-Weeks Gantt Chart

Software Development Life Cycle (SDLC) Model

Model Adopted: Agile Methodology

We chose Agile Methodology because the Neighborhood Watch Incident System requires continuous collaboration with community members, barangay officials, and law enforcement. Agile allows the system to be developed in iterative sprints, enabling quick feedback, rapid bug fixing, and adjustments to meet user needs.

• Flexibility – Supports feature changes such as SMS alerts, geospatial mapping, or anonymous reporting.

 • Collaboration – Continuous interaction ensures the system addresses real community safety challenges.

 • Faster Delivery – Working features are delivered incrementally for early testing.

• Risk Reduction – Sprint reviews detect issues early, reducing chances of failure.

Republic of the Philippines
Sorsogon State University
COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Bachelor of Science in Computer Science
Zone-8, Bulan, Sorsogon
Tel. No.; 056 311-9800; Email address: bscs_pc@sorsu.edu.ph

BAGONG PILIPINAS

16-Weeks Gantt Chart

Neighborhood Watch Incident System : A Digital Digital Platform for Safer Community

Task	February	March	April	May	June	September	October	November	December	January
	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5	W1 W2 W3 W4 W5
Planning Phase										
Define project objectives & scope										
Identify stakeholders (community, barangay officials, police)										
Conduct feasibility study										
Gather initial requirements										
Draft project charter & approval										
Analysis & System Design Phase										
Create detailed system requirements										
Prepare use case diagrams and flowcharts										
Design system architecture (frontend, backend, database)										
Design user interface mockups										
Validate design with stakeholders										
Implementation Phase (Development)										
Develop user registration & login module										
Develop incident reporting module (with image/video upload)										
Develop notification/alert system (SMS/email alerts)										
Develop community forum/discussion board										
Develop admin dashboard (incident monitoring & analytics)										
Integrate database (user profiles, incident logs)										
Implement security features (authentication, encryption)										
Testing Phase										
Perform unit testing (modules)										
Conduct integration testing (modules working together)										
Conduct system testing (performance & reliability)										
Conduct integration testing (modules working together)										
User acceptance testing (with community representatives)										
Bug fixing & refinement										
Deployment Phase										
Prepare deployment plan & documentation										
Deploy system on hosting/server										
Configure domain & security (SSL, backups)										
Train users (officials, community)										
Official system launch										
Maintenance Phase										
Continuous monitoring of incident reports										
Provide technical support										
Release updates & new features										
Provide technical support										