Functional Requirements:

1. Animal Movement Monitoring

Surveillance system for the observation of all movements in

radiotransmitters in animals inhabiting the park;

animals and tracking their locations down to 0.1 s individually

2. Fence Monitor & Alarm System

System — The system should sense any damage done to the

the animal, and start sounding alarm when the ()); [electric]] fences collapse.

necessary.

3. Veterinary Notification

Notify the veterinary staff - The system should notify the vet staff;\*/

(via a beep or alarm) when any of the animals are wounded.

Non-Functional Requirements:

1. Performance

Efficiency — the system must calculate location for each wildlife within 0.1 seconds,

and should be monitored seconds by second to have the quick slew rate monitoring and response.

2. Usability

Ease of Usewhereby the interface is in menus ensuring that anyone can use it

staff in the control center.

3. Reliability

Reliable: The system should be rigorously tested before it is delivered to ensure that the user can trust them.

since dangerous animals are being watched. It needs to work, and with no faults especially in.

fences broke and animal escapes occurred.

----

Part B

In this software development life cycle the most suitable sd1 would be Incremental Model.

cycle. In this post, I will explain why and how.

Why the Incremental Model

Model?

?

1. Complexity

Complexity: the system uses many as well as crucial functions (tracking animals movement, alert alarms, veterinary alerts and so on.) Building the

end to end system (like Waterfall Model) may involve risks and

inability to satisfy demand.

2. Iterative T

Testing

esting and Feedback

Comment: testing and Follows The Incremental Model

because of feedback on each increment, critical in a safety-critical system

which requires improvements to be made on a continuous basis in order for the

This application monitors dangerious animals.

3. Flexibility

Flexibility: The changes can be done from the development as and when required at any time.

For example, improvements can be made based on feedback from initial increments of the system.

is taken prior to the full thing being rolled out.

4. Time Sensitivity

Practicality: A December opening for the park is planned, and the system

must be ready by then. Incremental development of the system that delivers it early

core functionality (such as almost the basic tracking of mammals) and further enhance any for additional

features (like User Interface, Alarms and Veterinary Alert) in the following ones

increments.

Incremental Model Phases

Model::

1. Requirement Analysis

Analysis:

• Document features that will and won´t be supported by the system

functional requirements.

——Split the requirements across various modules or components which are interdependent,

be developed incrementally.

2. Design

Design:

For every iteration, develop the architectural design…

Modules/Services being Designed For example, start

to help with animal movement tracking and position calculations.

module first.

3. Progress (First Increment)

Increment):

• Implement the basics like receiving\_signals from

the transmitters, calculating the positions of animals and displaying these on a map.

• Test even this increment for accuracy and performance.

4. Testing

esting:

• Test the increment for every one you have developed to see it fulfills all

the requirements. • Each component of this tool is being tested due to dangers of these animals detected by the system,

Fence breaths or other functional failures not to happen so必须防呢队因———— Imperialistic wild ass dunk.

alarm triggers.

5. Delivery of First Increment

Increment:

Why wait: • When the base features of system are done in first iteration

(like animal tracking and fence monitoring) This data is passed on to it client.

input from the stakeholders.

- produce partially working systems to satisfy urgent demands, instead of

Additional increases are created.

6. Data — Repeat Steps for Next Increments

Increments:

• provided a unique features especially veterinary alert system, staff account→add functionality for sure and we have our Veterinary Alert System • useraccessible area list→ add feature \*-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------\*?

fence damage, and the alarm system would clarify in future control menus.

increments.

· Tests each increment, deliver it to the customer and receive feedback ·

7. Integration and Final T

Testing

esting:

• Once all increments are developed and tested separately, these must be better integrated

them into the final system.

• Conduct End- to-end system testing both manually and via CI Pipelines • Comprehensive 360 degree Product Testing as part of the workflow

out the fail-safes and alarms in conjunction as they turn out to be

Semi til avsystem med farliga djur

8. Deployment and Maintenance

Maintenance:

Once the whole system is integrated, deploy it for park usage

[Enter Seed Node] Maintain system, Operating changes on behalf of user

feedback or necessary fixes.

---

Reason for selecting the Incremental Model

Model::

• Reduced Risk

Creating the system one step at a time reduces your risk of building something no-one will use.

identifying large issues later in the development process (and a primary issue to our user)

For me, Erlang is all about safety ( safety-critical systems like monitoring dangerous animals ), fault-tolerance and distribution.

Early core features delivery

Advantages: The Incremental Model permits you to submit components

components of a system (e.g., core animal monitoring) are adopted sooner with minimum possible.MoveNext Backup from Last Inked Event:minorPotential minor backup to the last inked event ongoing production.consequenceTimeline ReviewlineReviewed?

system under construction As the image above clearly indicated the two systems have identical functions are operational.

• Flexibility for Changes

Flexibility: The model accepts feedback and allows for adaptations in followingincrements, essential for a comprehensive system like this.

By experiencing real-time results, some may even find they need testing along the way and possible adjustments.

• Time Management

Management: The park has a hard open date so delivering an

provide functional increments partial the system opperational

And there are more features in active development.

The Incremental Model encapsulates the flexibility of incorporating changes at minimal cost, reduces risk and is able to demonstrate a functional aspect in every cycle.

These are all processes of continuous improvement, testing and refining that are required for a system as crucial but underdeveloped as the security+.

safety-focused project.

---

Summary of Why W

Waterfall

Waterfall and Reuse-Oriented Development Models

Not Fit:

1. Waterfall Model

Model:

• Rigid structure

FORMAT: Given that the system features safety-critical

issue and changes during development, components

would cost prohibitive to resolve Waterfall style. This model does

no slack is available and testing happens only at the endangible

far too dicey for a project this important.

• Late integration

interoperability, data exchange: Animal monitoring software must work

smoothly — and can fill any holes in your integration process during later stage integrations ]);

delay the opening of same respectively both for you to miss that deadline.

• Does not fit to requirement changes

requirements: If the requirements change during

feedback from initial testing) the development.

The Waterfall Model would not handle these needs and changes easily.

2. Reuse-Oriented Model

Model:

Relying on external dependencies

component: This model is pretty dependent on consists of the

lack of reusable component -A reusable components may not exists for a same

Specialized system (eg: monitoring dangerous animals) Customizing

the SFF-related savings from shared, reusable components can be offset by equivalent effort to make those processes and tools uniform across the enterprise

Reuse and its time and cost benefits.

• Integration issues

problem: Compatibility related to the reusability of components from other sources may incluCargo Cult Integration issues By necessitating up-to-the-moment data sets

monitoring and safety measures, these struggles could be incorporated

critical delays.

• Limited flexibility

flexible: Although this model could help reduce time and spending if the

wrong components are located, and misses the venturing flexibility.

the park monitoring system, and custom requirements of the part

Incremental Model is Best Suitable:

The Incremental Model is a blend of flexibility and risk minimization, both executed in parallel with the infrastructure for an otherwise finished component.

delivery. An iterative approach ensures delivery of essential features quickly, also testing them continuously.

making it the perfect fit for this project due to its nimbability and agility.

---

Part C

Before the computer monitoring system for park

problems, and when it comes to site hosting dangerous animals this is also a potentially bigger problem. I came up with six potential issues:

1. ◦ Incomplete or ambiguous requirements

Requirements::

• Stakeholder Confusion : The stake holders may not know exactly what they want out of the • No Training and SPOF(floor which plays a crucial role).

system. Generally speaking, these tools will try to use a few fancy buzzwords to explain how the monitoring whatchamacallit works (e. g.: multi-tenancy etc.).

or alert systems should work. This can result in incomplete requirements that require to be-built.

It requires clarification and can disrupt development timelines.

2. Diverse Stakeholder Needs

Needs::

• Multiple stake holders are a part of the project, park managers, game wardens

veterinarians, and safety officials (and maybe tourists) as well. Each group may have

Wanting to meet them all, but they are going in different directions and have other needs-through reconciling the mom of it all.

in system-level design. 3. Lack of T

Technical

Knowledge Transfer From Stakeholders

Stakeholders::

• The system may not be understood by many stakeholders.

( Radio Signal Broadcast, Serial Communication Form and Software Interface)

design). When they do, they may not be able to articulate what it is that would meet their

suitable for the plant development group (causing conversation gaps and).

misunderstandings.

4. Complex Safety Requirements

Requirements::

• Because the dangerous animals in this system must be monitored, security measures •adopted to meet these specifications differ.

it is absolutely necessary to be very detailed and accurate. Fine-tuning the parameters of alarms.

monitoring with fences, and emergency protocols can be challenging in areas which have not

room for error. A misunderstanding of safety protocols, literally for a moment…

safety-affecting system failures.

5. Evolving Requirements

Requirements::

Parks Management : As it is completed or as the parks are developed by system development, stakeholders may

recognize that some things about the system have got to give. For example, they might

choose to add animals or new attractions for tourists, having an impact on the

monitoring system. The requirements can change so much while the software is being designed that it may seem like impossibly hard to arrive at an End Product as worthy of your product name

system specifications.

6. Feasibility Constraints

Constraints::

• Balancing the costs with Rs 6 millionbudget of and timeline for the project.

might be difficult due to some constraints ( so close deadline in December ). Stakeholders may have

very high system features>=most resource-heavy record and data transcribing)>=detailed animal tracking for a species which has not been seen or recorded in \_ years

(like sending alerts [] or complex alarms), but those requirements may be harder to address in

the given cost and timeline. Prioritizing requirements in

It can be tough with these constraints.

To handle such challenges, it is important to regularly communicate with the parents/ care givers of.

Get input from stakeholders, do iterative requirements gathering and use techniques like prototypes.

clarify the system's needs, and also to come up with new user stories.