Name: Phung Xuan Tung

ID: 103792054

Group 2

Scenario Overview

Our development team embarked on creating a car sales website for Top Deal Auto. The project aimed to deliver a user-friendly platform that maximizes business value while adhering to timelines and managing risks. However, as development progressed, several factors deviated from our initial expectations.

Original Estimation (From 6.1C)

F1. "Allow user to see selling cars"

1. **UI Design for Selling Cars Page (Assigned to Phung Xuan Tung):**
    - **Task: Design wireframes for the page displaying selling cars.**
    - **Timeframe: 120 minutes.**
2. **Backend Setup (Assigned to Tran Tuan Nam):**
    - **Task: Determine data structure for selling cars and set up backend routes/controllers.**
    - **Timeframe: 120 minutes.**
3. **Frontend Setup (Assigned to Phung Xuan Tung):**
    - **Task: Implement frontend components and integrate with backend API endpoints.**
    - **Timeframe: 90 minutes.**
4. **Filtering and Sorting (Assigned to Tran Tuan Nam):**
    - **Task: Implement functionality for users to filter and sort selling cars.**
    - **Timeframe: 60 minutes.**
5. **Pagination (Assigned to Phung Xuan Tung):**
    - **Task: Implement pagination for the selling cars page.**

- **Timeframe: 30 minutes.**

6. **Car Detail View (Assigned to Phung Xuan Tung):**
    - **Task: Implement functionality for detailed car view and layout design.**
    - **Timeframe: 90 minutes.**
7. **Integration with User Authentication (Assigned to Tran Tuan Nam):**
    - **Task: Ensure authenticated access and implement redirects.**
    - **Timeframe: 30 minutes.**
8. **Testing (Assigned to Tran Tuan Nam):**
    - **Task: Write unit and integration tests.**
    - **Timeframe: 120 minutes.**
9. **Bug Fixes and Refactoring (Assigned to Tran Tuan Nam):**
    - **Task: Address bugs and refactor code.**
    - **Timeframe: 60 minutes.**
10. **Documentation (Assigned to Pham Duc Linh):**
    - **Task: Document setup instructions and user guide.**
    - **Timeframe: 60 minutes.**

Unexpected Twists

As development progressed, several factors deviated from our initial expectations. Let's analyze each activity and discuss the reasons for the deviations:

1. UI Design for Selling Cars Page (Phung Xuan Tung)
    - Estimated Time: 120 minutes
    - Actual Time: 150 minutes (25% deviation)
    - Reasons for Deviation:
        - Complex Requirements: The initial wireframe design encountered unexpected complexities due to additional features requested by stakeholders. These modifications extended the design phase.
        - Iteration and Feedback: Iterative design cycles and feedback from team members and stakeholders contributed to the extra time spent.
    - What could be done:
        - Clear Requirements: Ensure thorough requirements gathering upfront to minimize surprises during design.

        - Regular Checkpoints: Frequent design reviews and checkpoints can prevent scope creep.

Note: The wireframe design process took longer due to additional iterations and feedback from stakeholders. While creativity is essential, we should allocate more time for design refinement in future projects.

2. Backend Setup (Tran Tuan Nam)
    - Estimated Time: 120 minutes
    - Actual Time: 130 minutes (8.3% deviation)
    - Reasons for success:
        - Framework Learning Curve: Setting up backend routes/controllers required familiarizing with a new framework, leading to a slight delay.
        - Optimization: Spending additional time on optimization and database schema design contributed to the deviation.
    - Actions for future Success:
        - Skill Enhancement: Continued learning and practice with the chosen framework will improve efficiency.
        - Predefined Templates: Utilize predefined templates or boilerplate code for common tasks.

Note: Tran Tuan Nam efficiently determined the data structure and leveraged existing frameworks. The accuracy here demonstrates the benefits of careful planning and familiarity with backend technologies.

3. Frontend Setup (Phung Xuan Tung)
    - Estimated Time: 90 minutes
    - Actual Time: 100 minutes (11.1% deviation)
    - Reasons for Deviation:
        - Integration Challenges: Integrating frontend components with backend API endpoints required additional troubleshooting.
        - Component Reusability: Reusable components took longer to set up initially but will save time in subsequent features.
    - What could be done:
        - Component Libraries: Leverage existing component libraries to streamline integration.
        - Collaboration: Collaborate closely with backend developers during integration.

Note: My modular approach to frontend integration paid off. Breaking down the task into manageable stages allowed for quicker initial setup.

4. Filtering and Sorting (Tran Tuan Nam)
    - Estimated Time: 60 minutes

- Actual Time: 55 minutes (8.3% deviation)
- Reasons for success:
  - Existing Libraries: Leveraging existing sorting/filtering libraries allowed faster implementation.
  - Focused Approach: A focused session on this specific feature minimized deviations.
- Action for future Success:
  - Reuse Libraries: Continue using established libraries for common functionalities.
  - Task Segmentation: Break down tasks into smaller, focused sessions.

Note: Despite using existing libraries, the implementation required some adjustments. Future estimations should account for potential complexities even in seemingly straightforward tasks.

5. Pagination (Phung Xuan Tung)
   - Estimated Time: 30 minutes
   - Actual Time: 35 minutes (16.7% deviation)
   - Reasons for Deviation:
     - Framework Tools: Utilizing built-in pagination tools was straightforward but required minor adjustments.
     - Testing and Validation: Additional time was spent validating pagination behavior.
   - What could be done:
     - Test Scenarios: Define test scenarios for pagination during estimation.
     - Documentation: Document pagination behavior for future reference.

   Note: The quick implementation aligns with the standard nature of pagination features. Well-defined requirements contributed to accuracy.

6. Car Detail View (Phung Xuan Tung)
   - Estimated Time: 90 minutes
   - Actual Time: 85 minutes (5.6% deviation)
   - Reasons for good prediction:
     - Template Adaptation: Adapting existing components for detail views streamlined development.
     - Focused Implementation: A focused session allowed efficient coding.
   - Future Success:
     - Component Reuse: Continue leveraging existing components where applicable.

        - Code Reviews: Regular code reviews can catch any deviations early.
7. Integration with User Authentication (Tran Tuan Nam)
    - Estimated Time: 30 minutes
    - Actual Time: 40 minutes (33.3% deviation)
    - Reasons for Deviation:
        - Middleware Configuration: Setting up authentication middleware took longer than anticipated.
        - Redirect Handling: Implementing redirects required additional time.
    - What could be done:
        - Authentication Templates: Use predefined authentication templates for faster setup.
        - Prioritize Security: Balance speed with security considerations.

    Note: Authentication flows were straightforward, but handling redirects and ensuring security took a bit longer. Future estimations should consider these nuances.

8. Testing (Tran Tuan Nam)
    - Estimated Time: 120 minutes
    - Actual Time: 130 minutes (8.3% deviation)
    - Reasons for good prediction:
        - Test Suite Setup: Initial test suite setup and configuration contributed to the deviation.
        - Coverage: Comprehensive testing took slightly longer.
    - Action for Future Success:
        - Test Suite Setup: Shorten initial test suite setup and configuration contributed to the deviation.
        - Automated Testing: Invest in automated tests.

    Note: Automated test suites are valuable, but thorough testing requires attention to detail. The slight overrun emphasizes the importance of quality assurance.

9. Bug Fixes and Refactoring (Tran Tuan Nam):

    Estimated Time: 60 minutes.

    Actual Time: 55 minutes.

    Reasons for Accuracy:

    - Decent estimation: The initial estimate somewhat accounted for the complexity of certain bugs.
    - Some Historical Data: Thanks to some outside source of bug fix metrics, it was easier to predict accurately.

Action for future Success: We'll allocate more time for bug fixes in future sprints. Addressing initial bugs and refactoring took slightly longer than expected. Some issues required deeper investigation and adjustments.

Note: Addressing initial bugs and refactoring was efficient. However, more complex issues may arise later, so flexibility is crucial.

10. Documentation (Pham Duc Linh):

Estimated Time: 60 minutes.

Actual Time: 70 minutes.

Reasons for Accuracy:

- Experience: Pham Duc Linh had prior experience with documentation.
- Concise Content: Focused on essential information, avoiding unnecessary details.

Action for future Success: Continue leveraging expertise for documentation tasks. Documenting setup instructions and the user guide was completed efficiently.

Note: Generating documentation concurrently with development is a good practice. Clear instructions and user guides enhance project transparency.

Actual Total Effort vs. Original Estimation

Our original estimation aimed for a balanced approach, considering business value, development effort, and other factors. However, the actual total effort exceeded our expectations by more than 10%.

Our error margin fell outside the acceptable range. Here's how we can improve:

1. Refine Design Estimates: Allow additional time for design iterations and stakeholder feedback.
2. Balance Efficiency and Quality: Leverage existing frameworks but account for potential complexities.
3. Modular Approach: Break down tasks into manageable stages for quicker initial setup.
4. Consider Nuances: Even straightforward tasks (like authentication) may have hidden complexities.

5. Quality Assurance Matters: Allocate sufficient time for thorough testing.
6. Stay Flexible: Be prepared for more complex issues during bug fixes and refactoring.
7. Concurrent Documentation: Generate documentation alongside development to streamline the process.
8. Learn from Experience: Use historical data to refine estimates.
9. Collaborate: Share insights across team members to enhance accuracy.
10. Iterate: Regularly review and adjust estimates based on real-world experiences.

Conclusion

While our initial estimation framework was robust, external factors disrupted our plans. Moving forward, we'll refine our estimation techniques, maintain flexibility, and learn from deviations. By doing so, we'll achieve better alignment between estimates and actual effort, ensuring successful project deliveries.