As I put myself into the situation of the software engineer, here are the suggestions I researched and thought of for the improvement of the management for their future projects.

1. **Project Planning and Scope Management:**

Realistic Scheduling:
Ensure that the project timeline is realistic. This involves a thorough analysis of all tasks, dependencies, and potential risks. Use historical data from previous projects to estimate more accurately.

Milestones and Checkpoints:
Establish clear milestones and checkpoints. Regularly review progress against these milestones to identify any delays early.

2. Resource Management:

Adequate Staffing:
Ensure that the project is adequately staffed. If tasks are falling behind, it may indicate that additional resources are needed.

Skills Assessment:
Match tasks with team members' skills. Ensure that the right people are working on the right tasks to maximize efficiency and quality.

3. Agile Methodologies:

Iterative Development:
Use agile methodologies to break the project into smaller, manageable iterations or sprints. This allows for continuous integration and delivery, making it easier to identify and address issues early.

Regular Stand-ups and Reviews:
Hold daily stand-up meetings and regular sprint reviews to ensure everyone is on the same page and to identify blockers promptly.

4. Quality Assurance and Testing:

Continuous Testing:
Integrate testing into the development process from the beginning. Adopt practices like Test-Driven Development (TDD) and Behavior-Driven Development (BDD) to catch issues early.

Automated Testing:
Implement automated testing for regression, unit, and integration tests. This helps in quickly identifying bugs as new features are developed.

5. Risk Management:

Risk Identification and Mitigation:
Identify potential risks early in the project and develop mitigation plans. Regularly revisit these risks to ensure they are being managed effectively.

Contingency Planning:
Have a contingency plan for critical path tasks. If key tasks are delayed, have a backup plan to keep the project on track.

6. Communication and Collaboration:

Transparent Communication:
Foster a culture of open and transparent communication. Ensure that any issues, delays, or changes are communicated promptly to all stakeholders.

Collaborative Tools:
Utilize collaborative tools like JIRA, Trello, or Asana for task management and Slack or Microsoft Teams for communication. This ensures everyone has visibility into the project's progress and can collaborate effectively.

7. Bug Management:

Prioritize Bug Fixes:
Prioritize bug fixes based on their severity and impact on the project. Critical bugs should be addressed immediately to avoid cascading issues.

Bug Triage Meetings:
Hold regular bug triage meetings to review and assign bugs to the appropriate team members. Ensure that bugs are tracked and resolved in a timely manner.

8. Post-Mortem Analysis:

Retrospectives:
Conduct regular retrospectives to evaluate what went well and what didn't. Learn from these sessions to continuously improve your processes.

Root Cause Analysis:
For any significant delays or issues, perform a root cause analysis to understand why they occurred and how they can be prevented in the future.

By addressing these areas, their project management processes can be significantly improved, reducing the likelihood of incomplete tasks, unresolved bugs, and delayed testing. This will help ensure that future projects are delivered on time and with the expected quality.