Scenario 1:

**Diagnosis:** The daily stand-up is not being conducted efficiently, leading to excessively long meetings. This defeats the purpose of a short, focused check-in and impedes team productivity. The extended discussions on dependencies indicate a lack of prior resolution or planning for such issues.

**Remedy:** Enforce the 15-minute time-box for the daily stand-up. Any discussions that require extensive conversation should be flagged and addressed in a separate meeting with only the relevant parties involved. Additionally, consider implementing a pre-stand-up coordination or a separate technical discussion meeting to address complex dependencies.

Scenario 2:

**Diagnosis:** The development team is not practicing self-organization and is relying on the Scrum Master for task assignment, which is contrary to Scrum principles.

**Remedy:** Reinforce the importance of self-organization within the team. The Scrum Master should coach the team on how to manage and assign tasks among themselves. During sprint planning, ensure the team discusses and agrees on who will take on specific tasks or how tasks will be claimed throughout the sprint. Regularly remind the team of their responsibility to manage their own work.

Scenario 3:

**Diagnosis:** There is a communication barrier between the development team and the Product Owner, with the Scrum Master acting as an unnecessary intermediary.

**Remedy:** Encourage direct communication between the development team and the Product Owner. The Scrum Master should facilitate this by organizing regular meetings where developers can discuss tasks and requirements directly with the Product Owner. Educate the team on the importance of direct collaboration and make it clear that approaching the Product Owner directly is acceptable and encouraged.

Scenario 4:

**Diagnosis:** The team is struggling to adopt the new Agile Application Lifecycle Management tool, which is hindering their productivity and effectiveness.

**Remedy:** Provide comprehensive training sessions and continuous support for the team to get comfortable with the new tool. Identify key team members who can become tool experts and act as internal trainers or support for their peers. Additionally, consider simplifying the use of the tool by focusing on its most essential features initially and gradually introducing more advanced functionalities as the team becomes more proficient. Regularly collect feedback and address any usability issues the team encounters.