1. The six key roles mentioned by feature driven development are as follows:

Domain Experts:

- **Skills and Knowledge Needed:** Domain experts serve as vital links between the technical team and end-users, necessitating a profound understanding and expertise in the relevant industry or domain. Domain experts in this context would be individuals with a deep understanding of transportation regulations, pedestrian safety, and vehicle communication standards. They should be familiar with the legal and ethical considerations surrounding the use of car horns in various situations. Effective communication and analytical skills are essential for conveying user requirements to the development team. During the design phase, they conduct comprehensive domain walkthroughs. Additionally, familiarity with software development cycles and terminology facilitates collaboration with developers, enabling them to contribute to creating user stories and acceptance criteria.

- **Experience Required:** Substantial experience within the domain is crucial for accurately interpreting user needs, typically spanning several years of practical industry experience. Domain experts should have practical experience in the automotive industry, particularly in areas related to vehicle safety and regulation compliance. They should ideally have several years of experience working with autonomous vehicle technologies and understanding the intricacies of traffic dynamics and pedestrian behavior.While domain experts primarily focus on user requirements, those with a background in development can offer valuable insights and guidance.

Project Manager:

- **Skills and Knowledge Needed:** Project managers oversee the entire software development lifecycle, requiring strong organizational and leadership skills to coordinate team activities and manage timelines effectively. Project managers for this task should possess a strong understanding of software development processes and automotive project management. Proficiency in project management methodologies and tools is essential, along with the ability to provide progress reports to clients and make critical decisions regarding budgeting and resource allocation. Knowledge of Agile methodologies or similar iterative approaches may be beneficial given the iterative nature of software development.

- **Experience Required:** Considerable experience in project management, particularly in software development or related fields, is necessary. They should ideally have extensive experience in the automotive industry. This typically involves managing projects of varying complexities over several years.

Chief Architect:

- **Skills and Knowledge Needed:** Chief architects are responsible for designing the overall structure and architecture of the software, ensuring a clear understanding among team members and guiding the project through technical challenges. Profound knowledge of software design principles, system architecture, and scalability considerations is crucial, along with proficiency in relevant technologies and frameworks. During model development, they lead the modeling team and refine the object model as requirements evolve. For this project chief architects need to have expertise in designing software systems for autonomous vehicles, with a focus on real-time responsiveness and safety-critical functionalities. They should have a deep understanding of vehicle-to-pedestrian communication protocols and sensor fusion techniques.

- **Experience Required:** Chief architects should have extensive experience in software development for autonomous vehicles or similar safety-critical systems. In particular they should have experience in designing complex systems and model development typically acquired through many years of hands-on experience. They should have a proven track record of designing and implementing complex software architectures, ideally with experience in integrating new features into existing autonomous driving systems.

Development Manager:

- **Skills and Knowledge Needed:** Development managers oversee the development team, ensuring the timely delivery of high-quality software through effective leadership and communication. They collaborate with the chief programmer and architect to plan features and development sequences, providing guidance on technical challenges and ensuring alignment with project goals. They need to have the skills to guide the development team in implementing the new horn feature.

- **Experience Required:** Significant experience in software development and team management or leadership roles is essential, typically gained over several years in the industry. Development managers should have a background in software development, preferably with experience in embedded systems or automotive software.

Chief Programmer:

- **Skills and Knowledge Needed:** Chief programmers lead the feature team in implementing the software solution, requiring extensive knowledge of programming languages, development methodologies, and design patterns. They collaborate with the business to define feature lists and refine the object model during the design phase. Strong problem-solving skills are crucial for addressing complex technical challenges. For this project, chief programmers should have expertise in embedded systems and real-time software development. They need to have a deep understanding of vehicle control systems and sensor integration.

- **Experience Required:** Chief programmers should have a wealth of experience in software development, demonstrating a track record of delivering high-quality code and mentoring junior developers. For this project, they should have a significant amount of experience in software development for autonomous vehicles or similar embedded systems. They should have a track record of designing and implementing software features that meet stringent safety and reliability requirements.

Class Owners:

- **Skills and Knowledge Needed:** Class owners are developers responsible for implementing specific components of the horn feature, such as the horn control logic or the interface with sensor systems. They are responsible for designing and implementing individual classes and methods within the software application, requiring strong programming skills and a deep understanding of object-oriented design principles. Collaboration with team members and attention to detail are essential for maintaining conceptual integrity and developing unit tests.

- **Experience Required:** While not needing as much experience as senior team members, class owners should have a solid foundation in software development, typically gained through practical experience over several years. Class owners should preferably have knowledge of embedded systems or automotive software

2.

There are several advantages and disadvantages of segregating the software developer population by roles in feature driven development:

Advantages:

- **Specialization:** Role segregation enables individuals to focus on specific areas that are crucial for implementing the horn feature. For instance, domain experts can specialize in understanding traffic regulations and pedestrian behavior to determine appropriate horn usage scenarios. At the same time chief architects can concentrate on designing the system architecture to seamlessly integrate the horn functionality into the existing autonomous driving system without compromising safety or performance.

- **Efficiency:** With clear role definitions, team members can work more efficiently within their specialized domains. For example, development managers can coordinate the efforts of different teams. This ensures that each group focuses on its assigned tasks related to the horn feature implementation. This division of labor can lead to faster development cycles and timely delivery of the new feature.

- **Clear Accountability:** Role segregation promotes clear accountability for different aspects of the horn feature. For instance, chief programmers and class owners can be responsible for implementing the horn control logic and integrating it with sensor systems. Meanwhile, project managers ensure that timelines are met, and stakeholders are informed of progress. This clarity helps in tracking progress and addressing any issues that arise during development.

- **Effective Communication:** Clear role definitions facilitate effective communication within the development team and with stakeholders. For example, domain experts can communicate user requirements and safety considerations to the development team, while chief architects ensure that the technical implementation aligns with overall system requirements. Also, the chief architect can make sure that everyone understands the design. This communication ensures that the horn feature meets both user needs and technical standards.

- **Risk Mitigation:** Segregating roles helps mitigate the risks associated with developing safety critical features for autonomous vehicles. For instance, by having specialized roles, the team can ensure that each aspect of the horn feature is thoroughly addressed. This reduces the likelihood of errors or oversights that could compromise vehicle safety. Additionally, clear role definitions enable smoother transitions if team members need to be replaced or reassigned during the project.

Disadvantages:

- **Silos and Fragmentation:** Role segregation may lead to silos within the development team, where individuals focus solely on their assigned tasks without sufficient collaboration with others. For example, if domain experts and chief architects work in isolation, they may overlook important insights or constraints that could affect the implementation of the horn feature. This fragmentation can prevent innovation and result in solutions that are suboptimal.

- **Lack of Flexibility:** Strict role segregation may limit the team's ability to adapt to evolving project requirements or unexpected challenges. For instance, if a certain aspect of the horn feature requires input from multiple roles, rigid role definitions may slow down decision making and problem solving processes. Flexibility is important in order to address emerging issues and optimize the horn feature's functionality in real-world scenarios.

- **Communication Overhead:** While clear role definitions can improve communication, they may also introduce communication overhead. This in particular can happen if information needs to be communicated between multiple roles. For instance, if domain experts need to convey user feedback to development teams through project managers, that can lead to delays and misunderstandings. The streamlining of communication channels is essential to avoid unnecessary overhead.

- **Dependency Risks:** Specialization can create dependencies between roles, where certain tasks or decisions rely on the outputs of other roles. For example, if the implementation of the horn feature requires close coordination between chief programmers and class owners, delays or errors in one area may spill over to the other area and impact the entire development process. Managing dependencies effectively is crucial for maintaining project pace and mitigating risks.

- **Skill Set Gaps:** Role segregation may result in skill set gaps that could affect the quality of the horn feature implementation. For example, if there is a lack of collaboration between domain experts and development teams, the resulting solution may not fully address user needs or safety requirements.

In conclusion, while segregating the software developer population by roles can offer advantages such as specialization, efficiency, and clear accountability for adding a new horn feature to driverless cars, it also presents challenges such as silos, lack of flexibility, communication overhead, dependency risks, and skill set gaps. Finding the right balance between specialization and collaboration is essential for ensuring the successful implementation of the horn feature within the autonomous driving system.