1. Demographics

1.1. Years of professional experience

1.2. Years of SoS experience

1.3. Number/names of SoS projects/programs worked on

2. SoS Architecture Development

We are looking here for the architecture design process, not the documentation process (e.g., DoDAF).

For each question, ask interviewee specifically about software, as appropriate.

Try to be as specific as possible.

2.1. In your experience, how are SoS architectures defined? Is software treated separately or differently from the rest of the SoS?

2.2. What is the architecture design process? How do you bridge from SoS-level focus to system-level focus?

2.3. How are architecture design trade offs defined or framed? How do you balance SoS context with system context?

2.4. How are architecture design trade off decisions made? What factors (technical and non-technical) influence decisions? How do you balance SoS context with system context?

3. SoS Success Patterns and Challenges

Ask interviewee to refer to a specific system/SoS/program in answering these questions, and to be as specific as possible in describing the success or challenge.

We do not want to dig into root cause analysis – we are looking for the observable events.

When the interviewee identifies a system/program that “went well”, probe into how they defined success – e.g., on budget, on schedule, exceeded requirements, provided a platform that other programs could build upon, high benefit for cost, etc.

Challenges could include complying with standards, not complying with standards, silos, not-invented-here syndrome, legacy constraints, etc.?

3.1. In developing constituent systems for use in a SoS, was there a system/program that went well or was more successful than others? Why? (i.e. what evidence is there?)

3.2. What SoS-related challenges have you seen in the development of constituent systems for use in a SoS?

3.3. In the test, integration, and assurance of constituent systems into a SoS, was there a system/program that went well or was more successful than others? Why? (i.e. what evidence is there?)

3.4. What SoS-related challenges have you seen in the test, integration, and assurance of constituent systems into a SoS?

3.5. In the configuration and management of a SoS, was there a system/program that went well or was more successful than others? Why? (i.e. what evidence is there?)

3.6. What SoS-related challenges have you seen in the configuration and management of a SoS?

For the next two questions, include issues about evolution of a constituent system within the overall SoS.

3.7. In the post-deployment sustainment and evolution of a SoS, was there a system/program that went well or was more successful than others? Why? (i.e. what evidence is there?)

3.8. What SoS-related challenges have you seen in the post-deployment sustainment and evolution of a SoS?

4. Solution Constraints

“Solution” is used in a very general sense here, and refers to something that would address one of the *challenges* identified above.

Constraints can be technical, organization, governance, policy, regulatory, incentive, doctrine, or other cause.

4.1. What constraints do you see for a solution that improves the development of constituent systems for use in a SoS?

4.2. What constraints do you see for a solution that improves the test, integration, and assurance of constituent systems for use in a SoS?

4.3. What constraints do you see for a solution that improves the configuration and management of a SoS?

4.4. What constraints do you see for a solution that improves the post-deployment sustainment and evolution of a SoS?