**📄 Sheet: Business & Functional Understanding**

| **Key Questions** |
| --- |
| What is the primary purpose of this application? |
| Who are the end-users and what problems does it solve for them? |
| What are the core business processes this system supports? |
| What are the high-impact user journeys or scenarios? |
| Are there SLAs or compliance-related business rules to be enforced? |
| What is the regulatory or compliance scope (e.g., KYC, AML, PCI-DSS, GDPR)? |
| Any seasonal spikes or known peak load periods? |

**🧩 Sheet: System Architecture & Integration**

| **Key Questions** |
| --- |
| Can you share a high-level architecture diagram? |
| What are the major components/modules in the system? |
| How does the system integrate with other internal/external systems? |
| What APIs are exposed or consumed? |
| Is the system event-driven, synchronous, or batch-based? |
| How is data flow managed across services? |
| What are the key dependencies and how resilient are they? |

**🛠️ Sheet: Technical Stack & Infrastructure**

| **Key Questions** |
| --- |
| What is the technology stack for each layer? |
| What cloud/on-prem infrastructure is used? |
| Is it deployed on microservices, monolith, or hybrid? |
| What is the deployment pipeline (CI/CD)? |
| How is config and secrets management handled? |
| What are the logging, monitoring, and alerting tools used? |

**🧪 Sheet: Testing & Quality Assurance**

| **Key Questions** |
| --- |
| What is the test strategy – unit, integration, system, UAT? |
| Are there automated test suites? Which tools are used? |
| How is regression testing handled? |
| Who signs off on releases? |
| Are there performance/load tests? What are the benchmarks? |

**👥 Sheet: Teams & Responsibilities**

| **Key Questions** |
| --- |
| Who are the main stakeholders? |
| Who owns which modules or areas of functionality? |
| What are the support SLAs? Who handles production issues? |
| Is there a change management process for code, infrastructure, or data? |

**🚦 Sheet: Pain Points & Risk Areas**

| **Key Questions** |
| --- |
| What are the current challenges or bottlenecks? |
| Are there any frequent incidents or recurring bugs? |
| What’s the recovery strategy during outages or failures? |
| Are there technical debts or planned rewrites/refactors? |

**📚 Sheet: Documentation & Knowledge Sources**

| **Key Questions** |
| --- |
| Where is the documentation stored? |
| Are API specs, code repos, architecture diagrams up to date? |
| What are the common escalation points? |
| Are there any tribal knowledge or undocumented flows? |

## 10. Knowledge Sources

Documentation Repositories:

Code Repositories:

Tickets / Backlog:

Meeting Notes / Past KT:

## 11. Key Questions to Ask During KT

- What are the biggest pain points in the current system?

- Who are the top consumers of this application?

- Any major upcoming changes?

- What happens when the system goes down? (Recovery process)

- Who owns which modules?