SCRUM MASTER

🟢 Track 1: SIMPLE (Scrum Basics + Intro to SAFe Concepts)

Focus: Scrum fundamentals, basic Agile, and foundational SAFe awareness.

-- Scrum Framework Basics --

1. What is Scrum?
2. What are the 3 pillars of Scrum?
3. What are the 5 Scrum values?
4. What are the roles in Scrum?
5. What are Scrum artifacts?
6. What are Scrum events (ceremonies)?
7. What is the timebox for a Sprint?
8. Who owns the Product Backlog?
9. Who is responsible for removing impediments?
10. What is a cross-functional team?

-- Events & Artifacts --

1. What is the purpose of Sprint Planning?
2. What is the output of Sprint Planning?
3. What is a Daily Scrum?
4. Who attends the Daily Scrum?
5. What happens in a Sprint Review?
6. What is discussed in a Sprint Retrospective?
7. What is the Scrum Master's role in the Retrospective?
8. What is a Product Increment?
9. What is Definition of Done (DoD)?
10. What is Definition of Ready (DoR)?

-- Intro to SAFe Concepts --

1. What is SAFe (Scaled Agile Framework)?
2. What are the four levels of SAFe?
3. Who are the key roles in SAFe?
4. What is a Program Increment (PI)?
5. What is an Agile Release Train (ART)?
6. What is PI Planning?
7. What is the purpose of System Demo in SAFe?
8. What is the difference between Scrum and SAFe?
9. What is a Feature in SAFe?
10. What is an Epic in SAFe?

🟡 Track 2: MEDIUM (Applied Scrum + SAFe Practices)

Focus: Practical Scrum usage, metrics, team dynamics, and SAFe events/roles.

-- Scrum in Action --

1. How do you handle a Product Owner who frequently changes priorities?
2. How do you ensure the team meets the Definition of Done?
3. How do you handle unplanned work during a Sprint?
4. How do you measure Sprint success?
5. How do you motivate a demotivated team?
6. What's the difference between Velocity and Capacity?
7. How do you manage dependencies between teams?
8. How do you handle a team member who doesn't participate actively?
9. How do you handle conflicts between team members?
10. How do you coach a new Product Owner?

-- Metrics & Reporting --

1. What metrics do you track as a Scrum Master?
2. How do you use a Burndown/Burn-up chart?
3. How do you use retrospective outcomes for improvement?
4. How do you track and handle technical debt?
5. How do you use Velocity trends to predict delivery?

-- SAFe Applied Practices --

1. How do you facilitate PI Planning?
2. What is the Scrum Master's role in an ART?
3. What is the difference between Team PI Objectives and Program PI Objectives?
4. How do you handle dependencies across ARTs?
5. How do you track risks and impediments in SAFe?
6. What is the purpose of the System Demo?
7. How do you ensure alignment between multiple teams?
8. How do you coach teams in Lean-Agile mindset?
9. How do you handle stakeholders in SAFe?
10. What is Inspect & Adapt (I&A) in SAFe?

-- Team Collaboration & Conflict --

1. How do you facilitate cross-team collaboration?
2. How do you deal with scope creep during a Sprint?
3. How do you manage distributed or remote Agile teams?
4. How do you handle a team that resists Agile transformation?
5. How do you prioritize backlog items with multiple stakeholders?

🔴 Track 3: COMPLEX (Advanced, Enterprise-Level, SAFe & Leadership)

Focus: Scenario-based, scaling, metrics, servant leadership, SAFe enterprise.

-- Advanced Facilitation & Coaching --

1. Describe a time you resolved a major team conflict.
2. How do you coach senior management on Agile principles?
3. How do you deal with a micromanaging Product Owner?
4. How do you coach teams transitioning from Waterfall to Agile?
5. What techniques do you use to facilitate effective retrospectives?
6. How do you measure Agile maturity at a team or program level?
7. How do you build trust with stakeholders?
8. How do you handle a team consistently missing PI objectives?
9. How do you coach teams on self-organization?
10. How do you apply servant leadership in a complex enterprise?

-- SAFe Scaling & Program Management --

1. How do you handle dependencies across multiple ARTs?
2. How do you measure PI performance and success?
3. How do you facilitate risk management in SAFe?
4. What is the difference between PI Objectives, Features, and Epics?
5. How do you align team backlogs with program-level objectives?
6. How do you deal with distributed ARTs across multiple geographies?
7. How do you handle inter-team conflicts at the program level?
8. How do you track metrics like Predictability, NFR compliance, and Feature completion in SAFe?
9. How do you coach leadership on Lean Portfolio Management?
10. How do you implement continuous improvement at program and portfolio levels?

-- Metrics, Delivery & Leadership Challenges --

1. How do you report Agile metrics to executives?
2. How do you maintain quality while accelerating delivery?
3. How do you handle scope creep in large programs?
4. How do you manage remote or hybrid SAFe teams?
5. How do you foster innovation and Lean thinking in a conservative organization?
6. How do you balance delivery pressure and team well-being?
7. How do you implement continuous improvement across multiple teams?
8. How do you coach multiple Scrum Masters?
9. How do you handle changes to PI mid-cycle?
10. How do you recover from a failed PI or ART?

🌀 LEVEL 1

👋 -- Scrum Framework Basics -- 👇

📋1️⃣What is Scrum?

Definition:
Scrum is an **Agile framework** used to develop, deliver, and sustain complex products through **collaboration, adaptability, and iterative progress**.
It is lightweight, simple to understand but hard to master.

Purpose:
It helps teams deliver value **incrementally**, inspect and adapt continuously, and respond to change quickly.

Example:
A product team working on an insurance claims automation system delivers new features every 2 weeks, gathers user feedback, and improves the next sprint — instead of waiting months for a big release.

🧱2️⃣What are the 3 Pillars of Scrum?

The **three pillars** uphold every implementation of Scrum — they ensure *transparency, inspection, and adaptation.*

Pillar	Description	Example
Transparency	Everyone must see the same truth about work and progress.	Sprint Backlog visible to all team members and stakeholders.
Inspection	Regularly check progress toward the goal and look for deviations.	Daily Scrum helps inspect work and progress.
Adaptation	Adjust as soon as possible to minimize deviations.	If QA finds recurring defects, team adapts process to include automation or better testing.

💡 *These three work together — you can't adapt effectively without transparency and inspection.*

💎3️⃣ What are the 5 Scrum Values?

These values guide how Scrum team members behave and work together:

Value	Explanation	Example
Commitment	Everyone commits to the team goals.	Team commits to deliver 20 story points this sprint.
Focus	Work on the sprint goal without distractions.	Team avoids mid-sprint scope changes.
Openness	Be open about challenges, progress, and failures.	A developer informs early about a technical blocker.
Respect	Value each team member's contributions and opinions.	Dev respects QA's feedback on acceptance criteria.
Courage	Speak up about risks and make tough decisions.	Scrum Master escalates unrealistic timelines to leadership.

🧭 *Together, these create trust, accountability, and collaboration.*

👥4️⃣ What are the Roles in Scrum?

Scrum defines **three primary roles**, collectively called the *Scrum Team*.

Role	Responsibility	Key Focus
Product Owner (PO)	Owns the Product Backlog, defines priorities, ensures value delivery.	"What to build"
Scrum Master (SM)	Facilitates Scrum events, removes impediments, coaches the team in Agile practices.	"How to improve"
Developers (Team Members)	Build, test, and deliver increments of value each sprint.	"Build and deliver"

Example:
In a P&C Insurance platform,

- PO prioritizes features like policy renewals,
- Dev Team builds APIs and UI,
- Scrum Master ensures collaboration and blockers are resolved.

📦5️⃣ What are Scrum Artifacts?

Artifacts provide **transparency** and help inspect/adapt the work.

Artifact	Purpose	Example
Product Backlog	Ordered list of everything needed in the product. Owned by PO.	New policy issuance, claims automation, dashboard reports.
Sprint Backlog	Items selected for the current sprint + plan for delivering them.	8 stories chosen for Sprint 5.
Increment	Working, potentially shippable product output from a sprint.	A tested and deployed claims workflow module.

💡 Each artifact has a **commitment**:

- Product Backlog → **Product Goal**
- Sprint Backlog → **Sprint Goal**
- Increment → **Definition of Done**

🕑6️⃣ What are Scrum Events (Ceremonies)?

Scrum events are **time-boxed meetings** to create regularity and enable inspection/adaptation.

Event	Purpose	Timebox	Example
Sprint	The heartbeat of Scrum — a timebox to create an increment.	≤ 1 month (typically 2 weeks)	2-week development cycle.

Sprint Planning	Define the Sprint Goal and select backlog items to deliver.	8 hrs (for 1-month Sprint)	PO + team agree on 8 stories.
Daily Scrum (Stand-up)	Inspect progress, adapt daily plan.	15 mins/day	“What did I do? What will I do? Any blockers?”
Sprint Review	Demonstrate increment to stakeholders, get feedback.	4 hrs (for 1-month Sprint)	Showcase new claim submission feature.
Sprint Retrospective	Reflect and improve process.	3 hrs (for 1-month Sprint)	Decide to automate test deployment.

⏳7️⃣What is the Timebox for a Sprint?

- **Maximum**: 1 month (4 weeks)
- **Typical**: 2 weeks (most teams use this)
- It should be **consistent** throughout to maintain rhythm and predictability.

Example:
If a team works in 2-week sprints, every alternate Friday they hold a sprint review + retrospective before the next sprint planning.

📋8️⃣Who Owns the Product Backlog?

➡️ **Product Owner (PO)** owns and manages the **Product Backlog**.

Responsibilities:

- Prioritizing based on business value and ROI.
- Ensuring clarity and visibility of backlog items.
- Refining stories with acceptance criteria.

Example:
PO for a claims product prioritizes “Fraud detection” above “UI enhancements” since it brings higher business value.

🧱9️⃣ Who is Responsible for Removing Impediments?

➡️ **Scrum Master** is responsible for **removing impediments** that block team progress.

Examples of impediments:

- Technical issues (server downtime).
- Process blockers (waiting for approval).
- Resource constraints or external dependencies.

Example:
If developers can't access a test environment, the Scrum Master coordinates with the DevOps team to resolve it quickly.

🤝 🔟 What is a Cross-Functional Team?

A **cross-functional team** has all the **skills and expertise needed** to deliver a complete product increment — without relying on external teams.

Includes: Developers, Testers, UI/UX designers, Database engineers, etc.

Example:
A P&C Insurance Scrum team:

- UI Developer (policy screen),
- Backend Developer (Guidewire API),
- QA (automation testing),
- BA (acceptance criteria clarification).
  Together they can build and deliver one working feature end-to-end.

💡 *Cross-functionality = self-sufficiency → faster delivery + higher ownership.*

👋 -- Events & Artifacts -- 👇

🧭1️⃣What is the Purpose of Sprint Planning?

Purpose:
Sprint Planning kicks off the Sprint. It defines **what can be delivered** in the upcoming Sprint and **how** that work will be achieved.

It aligns the **Product Owner (what)** and the **Development Team (how)** on a **Sprint Goal**.

Key Objectives:

1. Define the **Sprint Goal** — the "why" of the Sprint.
2. Select **Product Backlog Items (PBIs)** to deliver.
3. Create a **plan** for delivering them (task breakdown, capacity check).

Example:
For an Insurance Claims System:
Sprint Goal → "Enable policyholders to upload claim documents."
PBIs selected → UI upload component, API for file handling, validation logic, and test automation tasks.

📦2️⃣What is the Output of Sprint Planning?

The **main outputs** are:

1. **Sprint Goal** — a single, focused objective that gives direction.
2. **Sprint Backlog** — selected Product Backlog Items + plan to deliver them.
3. **Commitment** from the team to achieve the goal.

Example Output:

- Sprint Goal: "Improve claims submission accuracy."
- Sprint Backlog: 6 stories (validations, error messages, data mapping).
- Tasks: API updates, UI form edits, and test scripts.

💡 The Sprint Backlog is **owned by the Developers**, while the **Sprint Goal** is **shared by the whole team**.

⏱️3️⃣What is a Daily Scrum?

Definition:
A **15-minute daily meeting** where the Developers inspect progress toward the Sprint Goal and adapt their plan for the next 24 hours.

It promotes **transparency, communication, and quick adaptation**.

Typical Questions (not mandatory):

1. What did I do yesterday that helped meet the Sprint Goal?
2. What will I do today?
3. Are there any impediments?

Example:

- "Yesterday I fixed the validation bug."
- "Today I'll work on unit tests."
- "I'm blocked by missing test data."

💡 It's **not a status meeting for the Scrum Master** — it's for the **team's self-management**.

👥4️⃣Who Attends the Daily Scrum?

- **Mandatory:** Developers (core participants).
- **Optional:** Scrum Master and Product Owner (if needed, but they usually observe).

Scrum Master's Role: Ensure the event happens and stays within the timebox.
Product Owner's Role: Attend if needed to clarify backlog items, not to track status.

Example:
In a 7-member team (5 developers + 1 QA + 1 UI designer), all participate to align their day's work toward the Sprint Goal.

🎯5️⃣What Happens in a Sprint Review?

Purpose:
To inspect the **Increment** and adapt the **Product Backlog** based on feedback.
It's a **collaborative working session**, not just a demo.

Participants: Scrum Team + Stakeholders (business, users, management).

Agenda:

1. PO reviews Sprint Goal and completed items.
2. Team demos the Increment (working product).
3. Stakeholders give feedback.
4. Discuss next priorities and potential backlog changes.

Example:
The team demonstrates "Claims upload and fraud detection rules."
Stakeholders suggest adding "upload progress indicator" for next sprint.

Output:

- Updated Product Backlog.
- Alignment on next priorities.

🔁6️⃣ What is Discussed in a Sprint Retrospective?

Purpose:
To reflect on the last Sprint and identify **ways to improve** team collaboration, tools, and processes.

Typical Topics:

1. What went well?
2. What didn't go well?
3. What can we improve?

Example:

- ✅ Good: Early story refinement helped smooth execution.
- ⚠️ Bad: Test environment was unstable.
- 💡 Action: Automate test data refresh before every sprint.

Output:
A **list of actionable improvement items** for the next sprint.

🧑‍🏫7️⃣What is the Scrum Master's Role in the Retrospective?

Scrum Master facilitates the Retrospective to ensure it's:

- Constructive and blame-free.
- Focused on continuous improvement.
- Results in specific, actionable steps.

Responsibilities:

- Encourage open discussion.
- Protect the team from finger-pointing.
- Ensure action items are tracked in the next sprint.

Example:
SM uses techniques like "Start–Stop–Continue" or "Mad–Sad–Glad" to help the team reflect, and logs improvement tasks in JIRA.

🧱8️⃣What is a Product Increment?

Definition:
A **working, usable, and potentially shippable piece of software** created during the Sprint.

Every Sprint builds on previous Increments to form the final product.

Key Point:
An Increment must meet the **Definition of Done** before it's considered complete.

Example:
After Sprint 4, a policyholder can submit claims online (Increment).
After Sprint 5, they can also upload documents (next Increment).

💡 The Increment is the **sum of all completed work** meeting the DoD.

✅9️⃣What is Definition of Done (DoD)?

Definition:
A **formal checklist** that ensures work meets the required quality before it's released or demoed.

It brings **transparency and consistency** to the team's understanding of "done."

Example DoD Items:

- Code peer-reviewed
- Unit tested and integrated
- No critical defects
- Functional and regression tested
- Documented and deployed to QA

💡 *If a story doesn't meet DoD → it's not "Done" and doesn't count toward Velocity.*

🚀 🔟 What is Definition of Ready (DoR)?

Definition:
A **set of criteria** a story must meet before being accepted into a Sprint.
It ensures the story is well-understood, sized, and ready for development.

Example DoR Items:

- Clear Acceptance Criteria
- Dependencies identified
- UI mockups attached
- Story estimated
- Test data requirements identified

Purpose:
To prevent half-baked or unclear stories from entering a Sprint.

Example:
Before a "Create New Claim" story moves into a sprint:
✅ Acceptance criteria defined
✅ API dependency resolved
✅ Estimation done → Story is "Ready"

⚖️ **Comparison Table**

Aspect	Definition of Ready (DoR)	Acceptance Criteria (AC)	Definition of Done (DoD)
Purpose	Ensures the story is ready to start	Defines functional conditions for acceptance	Ensures the story is completely finished

Focus	Preparation	Functionality	Quality & completion
When Used	Before Sprint starts	During development & testing	After development/testing
Who Defines	Scrum Team	Product Owner (with QA/Dev input)	Scrum Team
Applies To	Every story before Sprint	Each individual story	Every story after Sprint
Example	“Story estimated and has AC defined.”	“User sees an error for invalid password.”	“Code reviewed, tested, and deployed.”

DoR ensures the team is not starting unclear or half-baked work.
Acceptance Criteria define how the Product Owner will validate the story.
DoD ensures consistency in quality and completeness before marking work done.

🧩 Summary Snapshot

Concept	Purpose	Owner/Participants	Output
Sprint Planning	Define what & how	PO + Dev + SM	Sprint Goal + Sprint Backlog
Daily Scrum	Align & adapt	Developers	Updated daily plan
Sprint Review	Inspect increment	Scrum Team + Stakeholders	Feedback + Updated backlog
Retrospective	Improve process	Scrum Team	Action items
Increment	Working product	Dev Team	Usable software
DoD	Define quality “done”	Team	Quality checklist
DoR	Define readiness to start	PO + Team	Prepared stories

👋 -- Intro to SAFe Concepts -- 👇

🧭1️⃣What is SAFe (Scaled Agile Framework)?

Definition:
SAFe (Scaled Agile Framework) is a **set of organizational and workflow patterns** designed to **scale Agile principles** beyond a single team — across multiple teams, departments, and portfolios in large enterprises.

Purpose:
To align strategy with execution, ensuring that all Agile teams deliver value in a synchronized and predictable way.

Key Principles (borrowed from Lean, Agile, and Systems Thinking):

- Deliver value continuously.
- Apply systems thinking.
- Build incrementally with fast learning cycles.
- Decentralize decision-making.

Example:
In a large insurance organization, multiple teams work on Guidewire PolicyCenter, ClaimCenter, and BillingCenter. SAFe ensures they all align under one shared business vision (e.g., “Digital Claims Transformation”).

🏗️2️⃣What are the Four Levels of SAFe?

SAFe has **four configuration levels** — each suited for different organizational sizes and complexities.

Level	Description	Key Roles
1. Team Level	Where Scrum/Kanban teams execute work (same as Scrum).	Scrum Master, Product Owner, Developers
2. Program Level (ART)	Multiple Agile Teams (5–12 teams) collaborate as one Agile Release Train (ART).	Release Train Engineer (RTE), Product Manager, System Architect

3. Large Solution Level	Coordinates multiple ARTs working on a large, complex solution.	Solution Train Engineer (STE), Solution Architect, Solution Management
4. Portfolio Level	Aligns enterprise strategy and funding with Agile execution.	Lean Portfolio Management (LPM), Epic Owners, Enterprise Architect

Simplified View:
👉 Team → Program → Large Solution → Portfolio

💡 *Most organizations implement Team + Program + Portfolio levels (known as "Full SAFe").*

👥3️⃣ Who are the Key Roles in SAFe?

Role	Responsibility
Product Owner (PO)	Owns Team Backlog, defines user stories, clarifies acceptance criteria.
Scrum Master	Facilitates Scrum at team level, removes impediments.
Release Train Engineer (RTE)	Chief Scrum Master for the Agile Release Train; ensures alignment and flow.
Product Manager (PM)	Owns Program Backlog, defines Features, aligns business priorities.
System Architect/Engineer	Defines technical architecture for ART.
Business Owner	Provides business value and approves PI objectives.
Epic Owner	Manages large-scale business initiatives at Portfolio level.
Lean Portfolio Manager (LPM)	Aligns strategy, funding, and portfolio execution.

🕑4️⃣ What is a Program Increment (PI)?

Definition:
A **Program Increment (PI)** is a **timebox** (typically **8–12 weeks**) during which an Agile Release Train (ART) delivers incremental value through multiple iterations (sprints).

Structure:

- Usually 5 iterations:
  → 4 development sprints + 1 Innovation & Planning (IP) sprint.

Purpose:

- Synchronize teams.
- Deliver integrated business value.
- Inspect and adapt regularly.

Example:
For a "Claims Digitization PI," 8 Scrum teams might deliver:

- API for claim submission
- Fraud detection ML model
- Dashboard for claim tracking
  over 10 weeks.

🚆5️⃣What is an Agile Release Train (ART)?

Definition:
An **ART** is a **long-lived team of Agile teams (typically 50–125 people)** that work together to deliver continuous value to the customer.

Key Points:

- Operates on a common PI cadence.
- Includes cross-functional teams (Dev, QA, Architects, UX, DevOps).
- Delivers Features (not just stories).
- Guided by roles like RTE, Product Manager, System Architect.

Example:
An ART in an insurance company might include:

- PolicyCenter team
- ClaimCenter team
- Integration team
  All aligned to deliver "End-to-End Policy Lifecycle Modernization."

🗓️6️⃣What is PI Planning?

Definition:
PI Planning is a **two-day, face-to-face (or virtual)** event held at the start of each Program Increment.

Purpose:

- Align all teams on **business vision**, **PI objectives**, and **dependencies**.
- Commit to delivering prioritized Features during the PI.

Agenda Includes:

1. **Business context** and vision from leadership.
2. **Team breakouts** to plan iterations.
3. **Risk identification** and dependency mapping.
4. **Final plan review** and **confidence vote**.

Output:

- Team PI Objectives
- Program Board (showing features, milestones, and dependencies)

Example:
During PI Planning, each team commits to Features like “Automated Claim Settlement” or “Fraud Detection Alerts,” and dependencies are mapped visually on a shared board.

🧩7️⃣What is the Purpose of System Demo in SAFe?

Definition:
A **System Demo** is held at the end of every iteration (or after each PI) to show the **integrated work of all teams in the ART**.

Purpose:

- Demonstrate progress toward PI objectives.
- Validate that features are working end-to-end.
- Gather feedback early from stakeholders and business owners.

Example:
All insurance teams demo together:

- UI → Policy submission screen
- Middleware → API integration
- Backend → Data storage
  This ensures all pieces fit and deliver customer value.

⚖️8️⃣ What is the Difference Between Scrum and SAFe?

Aspect	Scrum	SAFe (Scaled Agile Framework)
Scope	Single team	Multiple teams and portfolios
Cadence	Sprint (1–4 weeks)	Program Increment (8–12 weeks)
Focus	Delivering user stories	Delivering Features and Epics
Roles	PO, SM, Developers	Adds RTE, PM, System Architect, LPM, Epic Owner
Artifacts	Product & Sprint Backlog	Adds Program Backlog, Solution Intent, Portfolio Kanban
Planning	Sprint Planning	PI Planning
Objective	Team-level agility	Enterprise-wide alignment and value delivery

💡 *Scrum = team-level agility.*
SAFe = enterprise agility built on top of Scrum.

🌟9️⃣ What is a Feature in SAFe?

Definition:
A **Feature** is a **service or capability** that delivers value to the end user and can be completed within one PI by an ART.

Attributes:

- Described in the **Program Backlog**.
- Sized to be completed within a PI.
- Has **Acceptance Criteria** and **Benefit Hypothesis**.

Example:
Feature → “Policyholder can submit claims via mobile app.”

Acceptance Criteria → Uploads must support up to 10 MB files, secure authentication required.
Benefit Hypothesis → Improves claim submission rate by 30%.

🏗️ 🔟 What is an Epic in SAFe?

Definition:
An **Epic** is a **large, cross-cutting business initiative** that requires analysis, multiple PIs, and often multiple ARTs to complete.

Types of Epics:

- **Business Epic:** Delivers customer or business value.
- **Enabler Epic:** Supports architecture, compliance, or infrastructure needs.

Example:

- Business Epic: “Digital Transformation of Claims Experience.”
- Enabler Epic: “Implement Enterprise Data Lake for Analytics.”

Process:

1. Epic captured in **Portfolio Kanban**.
2. Reviewed through **Lean Business Case**.
3. Approved and decomposed into **Features** → later broken into **Stories**.

💡 *Epics drive strategic change; Features deliver tactical value.*

🔍 Summary Table

Concept	Description	Timeframe	Owner
SAFe	Framework for scaling Agile enterprise-wide	—	Enterprise/Agile Coaches
PI (Program Increment)	Timebox for delivering value	8–12 weeks	ART Teams
ART (Agile Release Train)	Team of Agile Teams delivering Features	Long-lived	RTE

PI Planning	2-day event to plan upcoming PI	Start of PI	All Teams
System Demo	Integrated demo of progress	End of Iteration/PI	Scrum Teams + Stakeholders
Feature	Value-delivering capability	Within 1 PI	Product Manager
Epic	Large-scale initiative across PIs	Multiple PIs	Epic Owner

🌀 LEVEL 2

👋 -- Scrum in Action -- 👇

1️⃣ How do you handle a Product Owner who frequently changes priorities?

Approach:

- **Empathize first** — understand *why* priorities are changing. It could be market shifts, stakeholder pressure, or lack of backlog refinement discipline.
- Educate the PO about **impact of mid-sprint changes** on sprint goals, predictability, and team morale.
- Introduce a **change control policy**: changes only allowed before Sprint Planning.
- Encourage use of **Product Backlog refinement sessions** to prioritize based on business value and team capacity.
- If absolutely critical changes arise mid-sprint, negotiate with the team to **cancel or re-plan the sprint** transparently.

Example:

> “In one project, the PO kept adding stories mid-sprint. I facilitated a discussion showing how our velocity dropped 30% due to context switching. Together we implemented a rule — all priority changes go through backlog grooming and get considered in the *next sprint*. Our predictability improved from 60% to 90%.”

2️⃣ How do you ensure the team meets the Definition of Done (DoD)?

Approach:

- Make sure **DoD is visible** and agreed by all team members (on a board or tool).
- During sprint planning, discuss DoD for each story.
- During daily scrums, check if tasks align with DoD.
- Encourage *peer reviews, automated testing, and acceptance testing* as part of the DoD.
- Ensure the PO accepts only stories meeting the DoD.

Example:

> "I embedded DoD criteria into our Jira workflow — no story could move to 'Done' without unit test and code review checks. This reduced rework and increased release quality."

3️⃣ How do you handle unplanned work during a Sprint?

Approach:

- **Classify the unplanned work** — is it critical (e.g., production defect) or can it wait?
- If critical, discuss with PO and team; re-negotiate Sprint scope.
- For recurring unplanned work, reserve a **buffer (10–15%)** in sprint planning.
- Encourage **better backlog refinement** to anticipate such work.

Example:

> "We had frequent production support tickets. We analyzed patterns, allocated 15% buffer for reactive tasks, and improved stability by creating preventive automation scripts."

4️⃣ How do you measure Sprint success?

Metrics to consider:

- **Sprint Goal Achievement** – Did we meet the agreed goal?
- **Team Velocity** – Was it stable and predictable?
- **Defect Leakage / Quality** – Was the delivered product high quality?
- **Customer Satisfaction** – Did we meet business expectations?
- **Team Happiness** – Are team members motivated and engaged?

Example:

> “For me, success isn’t just story points completed but whether we delivered value aligning with the Sprint Goal and improved predictability.”

5️⃣ How do you motivate a demotivated team?

Approach:

- **Listen first** – understand root causes (burnout, unclear goals, recognition, conflicts).
- Reconnect the team to the **‘why’** — customer impact, value of their work.
- Celebrate small wins (demo appreciation, internal kudos).
- Ensure a sustainable pace (avoid constant overtime).
- Encourage autonomy, ownership, and participation in decision-making.

Example:

> “A team was demotivated after a failed production release. I organized a retrospective focused on wins and learning, introduced ‘demo shout-outs,’ and improved visibility of their achievements to leadership. Morale improved significantly.”

6️⃣ What’s the difference between Velocity and Capacity?

Term	Definition	Example
Velocity	Average story points completed per sprint (team’s output).	Team completes ~40 SP per sprint.
Capacity	Total available working hours in a sprint (team’s potential input).	5 devs × 8 hrs × 10 days = 400 hrs capacity.

Relation:
Velocity = what’s *typically delivered* based on past data.
Capacity = *available bandwidth* for the current sprint (accounts for leaves, holidays, etc.)

7️⃣ How do you manage dependencies between teams?

Approach:

- Identify dependencies during **PI or Sprint Planning**.
- Use **dependency boards / Program boards** (especially in SAFe).

- Promote early communication between teams.
- For recurring dependencies, work with architects and PO to **decouple systems** or enable independent releases.
- Track and review dependency risks in Scrum of Scrums.

Example:

> "In a multi-team Guidewire implementation, we visualized dependencies via Jira Advanced Roadmaps and daily syncs. This helped us avoid blockers and deliver integrated features smoothly."

8️⃣ How do you handle a team member who doesn't participate actively?

Approach:

- Have a **1:1 private conversation** to understand the reason (shyness, lack of clarity, burnout).
- During ceremonies, encourage their input gently and acknowledge contributions.
- Pair them with a mentor or more active member.
- If behavioral, coach them on teamwork expectations.

Example:

> "A tester was quiet during standups. I had a 1:1 to understand his comfort level and realized he felt his updates were unimportant. We clarified expectations and later he began volunteering testing insights in retrospectives."

9️⃣ How do you handle conflicts between team members?

Approach:

- Act as a **neutral facilitator**.
- Address early — don't let issues simmer.
- Use the **"listen–clarify–align"** technique.
- Bring focus back to *shared goals*, not personal opinions.
- If unresolved, involve PO or HR as a last resort.

Example:

> "Two developers disagreed on code design. I facilitated a whiteboard session to evaluate pros/cons objectively. The team agreed on the best option, and both members appreciated the structured approach."

🔟 How do you coach a new Product Owner?

Approach:

- Educate them on **Scrum principles**, roles, and ceremonies.
- Coach on **backlog management**, prioritization by business value, and writing clear user stories.
- Encourage **regular stakeholder engagement** and feedback loops.
- Partner with them in **Sprint Reviews** and refinement to balance business vs. technical needs.

Example:

> "A new PO joined from a business background. I ran quick sessions on INVEST user story writing, backlog grooming, and setting Sprint Goals. Within 2 sprints, backlog clarity improved and dev team confidence increased."
>
> INVEST - Independent, Negotiable, Valuable, Estimatable, Small, Testable.

👋 -- Metrics & Reporting -- 👇

1️⃣ What metrics do you track as a Scrum Master?

As a Scrum Master, my goal is to **track metrics that improve transparency, delivery predictability, and team health — not to micromanage**.

Key Metrics:

Category	Metric	Purpose
Delivery	**Velocity**	Measure average story points delivered per sprint.
	Sprint Burndown	Track daily progress toward sprint goals.
	Release Burn-up	Visualize progress toward release scope completion.

Predictability	**Commitment Reliability** (% of committed vs. completed)	Helps assess planning accuracy.
Quality	**Defect Density / Leakage**	Measure defects caught post-sprint.
Flow	**Cycle Time / Lead Time**	Track how long it takes to deliver a user story.
Team Health	**Team Happiness / Engagement**	Regular feedback surveys or pulse checks.

Example:

> "I used velocity, sprint burndown, and cycle time trends to forecast release readiness and identify process bottlenecks. When cycle time spiked, we discovered delays in code reviews and addressed them via pairing."

2️⃣ How do you use a Burndown/Burn-up chart?

📉 Burndown Chart

- **Purpose:** Tracks *remaining work* (y-axis) vs. *time* (x-axis) within a sprint.
- Helps the team visualize whether they're on track to complete the sprint scope.
- The **ideal line** shows expected progress; the **actual line** shows real progress.

Use:

- Review during Daily Scrum to identify if progress is lagging.
- If the curve flattens — signals blockers, overcommitment, or unplanned work.

Example:

> "In one sprint, the burndown plateaued midweek. The team identified a dependency on another squad, escalated early, and resolved it before the end of sprint."

📈 Burn-up Chart

- **Purpose:** Shows *total scope* vs. *completed work* over time.
- Unlike burndown, it **highlights scope changes** clearly — you can see when scope increases or decreases.

- Useful for **release planning and forecasting**.

Example:

> "We used a burn-up chart at the release level to communicate progress to leadership. It showed that while team velocity was stable, scope creep was causing delays — prompting better backlog control."

3️⃣ How do you use retrospective outcomes for improvement?

Approach:

1. **Facilitate actionable retrospectives** using techniques like "Start-Stop-Continue," "5 Whys," or "Sailboat."
2. **Prioritize 1–2 improvement actions per sprint** (avoid overwhelming the team).
3. Track them in **Jira or a visible improvement board**.
4. Review action progress in subsequent retrospectives.

Example:

> "One retrospective revealed long QA cycles as a bottleneck. We implemented parallel testing and automated smoke tests. Within 3 sprints, cycle time dropped by 20%."

4️⃣ How do you track and handle technical debt?

Approach:

- Make technical debt **visible** — log it as items in the product backlog or a separate "Tech Debt" board.
- Estimate its impact (time, performance, risk).
- Negotiate with the PO to **allocate capacity (e.g., 10–20%)** each sprint for refactoring or cleanup.
- Track trend of technical debt using metrics like **Code Quality (SonarQube)**, **Defect Reopen Rate**, and **Velocity impact**.

Example:

> "We tracked technical debt stories in Jira labeled as 'tech-debt' and dedicated one story per sprint. Over time, our defect leakage reduced 25% and stability improved significantly."

5️⃣ How do you use Velocity trends to predict delivery?

Approach:

- Calculate **average team velocity** over the last 3–5 sprints.
- Use it to forecast how many story points can be completed in future sprints.
- Combine with **remaining backlog size** to estimate *how many sprints* are needed to complete a release.
- Use **velocity trend charts** to spot anomalies (sudden drops indicate impediments or scope churn).

Example:

> "Our average velocity was 45 points per sprint. With 180 points left in the release backlog, we projected 4 sprints to completion. Using velocity trends, we identified overcommitment in one sprint and adjusted scope early, improving predictability."

✅ Quick Summary Table

Area	Metric or Technique	Purpose	Example Use
Delivery	Velocity	Forecast sprint/release completion	Plan release timeline
Progress	Burndown Chart	Track daily sprint progress	Detect mid-sprint blockers
Scope Control	Burn-up Chart	Visualize done vs total scope	Identify scope creep
Improvement	Retrospective Actions	Continuous improvement	Reduce cycle time
Quality	Tech Debt Tracking	Maintain code quality	Allocate 15% capacity
Predictability	Velocity Trends	Predict delivery	Forecast 4 sprints ahead

👋 -- SAFe Applied Practices -- 👇

1️⃣How do you facilitate PI Planning?

Purpose:
PI (Program Increment) Planning is a **cadence-based, face-to-face event (typically 2 days)** that aligns all Agile teams on the **Program Increment goals** (usually 8–12 weeks).

Steps to Facilitate:

1. **Pre-PI Preparation**
    - Ensure **team backlogs are ready** (refined, prioritized, with dependencies identified).
    - Confirm **business context** and **vision briefing** by Product Management.
    - Prepare **capacity and velocity data** for each team.
2. **During PI Planning:**
    - Facilitate team breakout sessions — help teams draft **Team PI Objectives** and identify **dependencies and risks**.
    - Manage timeboxes, encourage collaboration, and ensure **dependencies are visualized on the Program Board**.
    - Support **confidence vote** (Fist of Five) and **risk ROAMing** (Resolve, Own, Accept, Mitigate).
3. **Post-Planning:**
    - Capture final Program Board, risks, and team objectives in the Agile tool (e.g., Jira Align, Aha!, Rally).
    - Communicate outcomes to stakeholders.

Example:

> “In our Guidewire ART, I helped 8 teams plan a PI by ensuring their backlogs were refined two weeks prior. During the event, I facilitated dependency mapping across PolicyCenter and ClaimCenter teams, leading to a more predictable integration cadence.”

2️⃣What is the Scrum Master’s role in an ART?

Role Summary:
The Scrum Master supports the **team within the ART (Agile Release Train)** to deliver value while aligning with ART objectives.

Responsibilities:

- Facilitate all team-level ceremonies (Sprint Planning, Review, Retrospective).
- Ensure the team participates actively in **PI Planning**, **System Demos**, and **Inspect & Adapt**.
- Help **remove impediments** and escalate to the RTE when needed.
- Support **flow of value** and foster **continuous improvement**.
- Coach the team on **Lean-Agile practices** and **SAFe principles**.

Example:

> "I acted as a bridge between my team and the RTE — ensuring our team's blockers were visible at the ART Sync and maintaining alignment with program priorities."

3️⃣ What is the difference between Team PI Objectives and Program PI Objectives?

Type	Description	Example
Team PI Objectives	Goals each Agile team commits to for the PI, aligned to program vision.	"Automate policy renewal notifications."
Program PI Objectives	The collective summary of all teams' PI Objectives that form the **Program-level goal**.	"Enhance policyholder self-service capabilities across all channels."

Key Point:
Program Objectives are the **aggregate** of all Team Objectives, refined by Product Management for executive visibility.

4️⃣ How do you handle dependencies across ARTs?

Approach:

1. Identify dependencies early during **PI pre-planning** and **PI Planning**.
2. Use a **Program Board** (for within an ART) and **Solution Train Board** (for cross-ART dependencies).
3. Schedule **Scrum of Scrums (SoS)** and **PO Syncs** to manage ongoing coordination.

4. Assign owners for each dependency and track them in a shared tool (e.g., Jira Align).
5. Encourage architectural alignment and decoupling wherever possible.

Example:

> "PolicyCenter had dependencies on BillingCenter for payment validation. We visualized them on the Program Board and set weekly cross-team syncs. This reduced dependency blockers by 40%."

5️⃣ How do you track risks and impediments in SAFe?

Tools and Techniques:

- Use **ROAM Board** during PI Planning and I&A:
  - **R**esolved – handled.
  - **O**wned – assigned to someone.
  - **A**ccepted – acknowledged but not acted upon.
  - **M**itigated – action plan defined.
- Maintain **impediment log** at team and ART level.
- Review and update risks in **ART Syncs** and **Scrum of Scrums**.

Example:

> "We tracked PI-level risks using a ROAM board in Miro and linked them to Jira issues for follow-up. Each was reviewed in ART Sync until closure."

6️⃣ What is the purpose of the System Demo?

Purpose:
To demonstrate the **integrated value** delivered by all teams in the ART for that iteration (or end of PI).

Key Points:

- Conducted at the end **of each iteration** (or PI).
- Focus on working, tested software (not slideware).
- Allows **stakeholders to see progress** and provide feedback early.
- Validates cross-team integration and end-to-end functionality.

Example:

> “Our System Demo after Sprint 3 showcased an integrated policy quote flow across all modules, which helped stakeholders validate usability before final release.”

7️⃣ How do you ensure alignment between multiple teams?

Approach:

- Regular **Scrum of Scrums** to synchronize progress and dependencies.
- **PO Syncs** for alignment of priorities and backlog refinement.
- Maintain a **shared Program Vision, PI Objectives, and Roadmap**.
- Encourage use of **Agile tools (Jira Align, Azure DevOps)** for transparency.
- Ensure all teams attend **System Demos** and **Inspect & Adapt sessions**.

Example:

> “We used joint refinement sessions and a shared roadmap to ensure all teams aligned to the customer journey rather than just their feature backlog.”

8️⃣ How do you coach teams in the Lean-Agile mindset?

Coaching Techniques:

- Conduct short **Lean-Agile principle sessions** (e.g., flow, value, pull system).
- Promote **continuous improvement and self-organization**.
- Encourage **data-driven decision-making** — using metrics like lead time, WIP.
- Lead by example — act as a **servant leader**, not a command controller.
- Foster a **culture of relentless improvement**.

Example:

> “I coached the team to visualize WIP limits on a Kanban board, reducing multitasking and improving throughput by 15%.”

9️⃣ How do you handle stakeholders in SAFe?

Approach:

- Identify key stakeholders early (business owners, PMs, system architects).
- Involve them in **PI Planning, System Demos, and I&A**.
- Maintain transparency via **burn-up charts, dashboards, and metrics**.

- Translate technical progress into **business value** language.
- Manage expectations by showing trade-offs and risks.

Example:

> "We created a PI dashboard summarizing feature progress and risks for business owners, which improved trust and decision-making."

🔟 What is Inspect & Adapt (I&A) in SAFe?

Purpose:
The **I&A workshop** is held at the end of each PI to **reflect, measure, and improve** at the program level.

Key Components:

1. **PI System Demo** – demonstrate PI outcomes across ARTs.
2. **Quantitative & Qualitative Review** – review metrics (predictability, quality, flow).
3. **Problem-Solving Workshop** – identify systemic issues and define improvement actions.

Outcome:

- Concrete **improvement backlog items** for the next PI.
- Enhanced **predictability and performance** for future increments.

Example:

> "In one I&A, defect leakage was identified as a major issue. We introduced static code analysis tools (SonarQube) and pair programming, which reduced defects by 30% in the next PI."

✅ Summary Table

Topic	Key Idea	Example Impact
PI Planning	Aligns all teams on common goals	Improved cross-team predictability
ART Role	Facilitate team, escalate impediments	Smooth dependency management

Team vs Program PI Objectives	Team = specific goals; Program = aggregate	Better executive visibility
Dependencies	Use Program Board, SoS	Reduced cross-team blockers
Risks	Track via ROAM	Early mitigation of PI-level issues
System Demo	Show integrated value	Early feedback loop
Alignment	SoS + PO Sync + Shared Vision	Cohesive delivery
Lean-Agile Coaching	Promote flow, self-organization	Faster delivery, happier teams
Stakeholders	Transparent communication	Business confidence
Inspect & Adapt	PI-level continuous improvement	Predictability improvement

👋 -- Team Collaboration & Conflict -- 👇

1️⃣How do you facilitate cross-team collaboration?

Approach:

- **Identify dependencies early**: During PI planning and backlog refinement, map cross-team dependencies visually (Program Board / Jira Align).
- **Regular synchronization**: Conduct **Scrum of Scrums (SoS)** for teams to communicate progress, blockers, and risks.
- **Shared ceremonies**: Encourage teams to participate in joint retrospectives, demos, or refinement sessions when objectives overlap.
- **Foster open communication**: Use collaboration tools (Slack, Teams, Confluence) to maintain transparency.
- **Encourage knowledge sharing**: Pairing sessions, guilds, or communities of practice to share domain expertise.

Example:

> “In a multi-ART insurance project, we had overlapping features between PolicyCenter and ClaimCenter teams. By visualizing dependencies and holding weekly SoS, blockers were resolved 50% faster, and delivery became predictable.”

2️⃣ How do you deal with scope creep during a Sprint?

Approach:

- **Educate stakeholders and PO** about **impact on sprint goal and team focus**.
- Use **sprint boundary principle**: changes during the sprint are deferred to the next planning session unless critical.
- **Negotiate trade-offs**: If a change is urgent, discuss what current stories need to be deprioritized.
- **Track scope changes**: Document mid-sprint scope changes for retrospective review and process improvement.

Example:

> "During a sprint, a critical regulatory requirement was requested. We evaluated its urgency with the PO, removed a lower-priority story, and successfully delivered the regulatory update without breaking the sprint goal."

3️⃣ How do you manage distributed or remote Agile teams?

Approach:

- **Time zone consideration**: Align key ceremonies in overlapping hours.
- **Use collaboration tools effectively**: Zoom, Miro, Jira, Confluence for real-time visibility.
- **Daily Scrum rigor**: Encourage concise stand-ups with clear updates.
- **Promote asynchronous communication**: For design discussions or knowledge sharing.
- **Foster team culture**: Virtual team-building, recognition, and transparency.

Example:

> "I managed three teams across the US and India. By using overlapping hours for sprint planning, rotating meeting times, and Miro boards for dependency visualization, delivery was synchronized and engagement increased."

4️⃣ How do you handle a team that resists Agile transformation?

Approach:

- **Listen to concerns**: Understand why the team resists (fear of change, lack of trust, past failures).
- **Educate and coach**: Show benefits of Agile principles with real examples.
- **Start small**: Implement incremental changes rather than full-scale immediate adoption.
- **Lead by example**: Demonstrate Agile mindset — transparency, feedback loops, and continuous improvement.
- **Celebrate small wins**: Highlight quick successes to build confidence.

Example:

> "A legacy IT team was hesitant to adopt Scrum. I started by coaching them on daily standups and short retrospectives. Within 3 sprints, they observed better visibility and quicker decision-making, which increased engagement."

5️⃣ How do you prioritize backlog items with multiple stakeholders?

Approach:

- **Use value-driven prioritization**: Rank items based on business value, risk, dependencies, and ROI.
- **Collaborative decision-making**: Conduct stakeholder workshops to clarify priorities.
- **Weighted Shortest Job First (WSJF)**: Common in SAFe to prioritize items with the highest cost of delay vs duration.
- **Transparency and trade-offs**: Document prioritization rationale for alignment and communication.
- **Continuous refinement**: Adjust backlog priorities regularly as business needs change.

Example:

> "In a P&C insurance project, multiple stakeholders requested conflicting features. We used WSJF scoring in PI planning, aligned priorities with business objectives, and communicated trade-offs transparently, which led to smoother delivery without conflicts."

✅ Quick Reference Table

Question	Key Actions	Example Impact
Cross-team collaboration	Program Board, Scrum of Scrums, shared ceremonies	Faster dependency resolution

Scope creep	Educate PO, negotiate trade-offs, defer non-critical changes	Sprint goal maintained
Distributed teams	Overlapping hours, collaboration tools, async communication	Improved engagement & delivery
Team resisting Agile	Listen, coach, start small, celebrate wins	Increased adoption & confidence
Backlog prioritization	Value-driven, WSJF, stakeholder alignment	Clear priorities, reduced conflicts

🌀 LEVEL 3

👋 -- Advanced Facilitation & Coaching -- 👇

1️⃣ Describe a time you resolved a major team conflict.

Approach:

- Listen to all parties neutrally.
- Identify the root cause (technical disagreement, communication gap, or personal issue).
- Facilitate a structured discussion focusing on **facts and goals**, not personalities.
- Guide the team to agree on a solution and follow up.

Example:

> "Two developers disagreed on a PolicyCenter integration design. I facilitated a whiteboard session to evaluate pros and cons objectively. The team agreed on a hybrid approach, and both members felt heard. Productivity improved, and blockers were cleared within a day."

2️⃣ How do you coach senior management on Agile principles?

Approach:

- Educate through **workshops and demos** showing business value and transparency benefits.

- Present metrics like **velocity, cycle time, predictability, and ROI**.
- Highlight **risks of command-and-control vs. benefits of servant leadership and self-organizing teams**.
- Share success stories and incremental wins to build trust.

Example:

> "I ran a session for executives showing how backlog prioritization improved claims processing. After seeing improved throughput and customer satisfaction metrics, leadership supported full Agile adoption."

3️⃣ How do you deal with a micromanaging Product Owner?

Approach:

- **Educate on Agile principles** — PO focuses on 'what' and 'why,' not 'how.'
- Use **clear sprint goals and Definition of Done (DoD)** to build confidence.
- Encourage **regular reviews and backlog grooming**, but maintain team autonomy.

Example:

> "A PO kept directing developers mid-sprint. I used sprint goals and metrics to show that allowing the team to self-organize improved predictability. After a few sprints, the PO trusted the team to deliver."

4️⃣ How do you coach teams transitioning from Waterfall to Agile?

Approach:

- Start with **training sessions** on Agile principles and Scrum/SAFe framework.
- Introduce **incremental changes** — start with stand-ups, retros, and small iterations.
- Use **visible metrics and quick wins** to demonstrate value.
- Continuously coach on **self-organization and collaboration**.

Example:

> "A 20-member legacy team struggled with iterations. We started with two-week sprints, stand-ups, and backlog grooming. Within 3 sprints, they delivered working increments consistently, and adoption grew naturally."

5️⃣ What techniques do you use to facilitate effective retrospectives?

Approach:

- Use **structured formats** like “Start-Stop-Continue,” “5 Whys,” or “Sailboat.”
- Encourage **participation from all team members** (silent brainstorming, dot-voting).
- Limit to **1–2 actionable items per sprint**.
- Track improvements and review in the next retro for accountability.

Example:

> “We used dot-voting to prioritize process pain points. One action to automate smoke tests reduced QA cycle by 20% and became a visible improvement for the team.”

6️⃣ How do you measure Agile maturity at a team or program level?

Metrics & Methods:

- **Team-level:** Velocity consistency, sprint goal achievement, lead time, WIP adherence, and retrospective action completion.
- **Program-level:** ART predictability, PI objective achievement, dependency resolution, and value delivery frequency.
- Use **Agile maturity assessment frameworks** or surveys to capture behaviors (e.g., self-organization, collaboration).

Example:

> “I conducted a maturity survey across 5 teams and found low adoption of WIP limits. Coaching them improved flow efficiency and predictability within 2 PIs.”

7️⃣ How do you build trust with stakeholders?

Approach:

- Maintain **transparency** with metrics and progress reports.
- Deliver **predictable, high-value increments**.
- Communicate **risks, dependencies, and trade-offs** early.
- Listen to stakeholder concerns and **act on feedback**.

Example:

> “By sharing PI burn-up charts and risks openly, business owners trusted the teams’ estimates, reducing escalations and improving engagement.”

8️⃣ How do you handle a team consistently missing PI objectives?

Approach:

- Identify **root causes**: scope overcommitment, dependencies, lack of skills, or process issues.
- Facilitate a **problem-solving session** in Inspect & Adapt (I&A).
- Adjust PI planning, remove bottlenecks, and coach teams on realistic commitments.
- Track improvement with metrics like **objective completion %, velocity trends, and confidence votes**.

Example:

> “Our ART missed 2 consecutive PI objectives due to dependency misalignment. We introduced joint refinement sessions and adjusted scope realistically. Next PI, 90% of objectives were met.”

9️⃣ How do you coach teams on self-organization?

Approach:

- Encourage **team decision-making** instead of directing.
- Promote ownership of **tasks, quality, and delivery**.
- Provide guidance on **collaboration, WIP management, and peer accountability**.
- Gradually step back and let the team solve problems independently.

Example:

> “I facilitated workshops where the team decided task assignments and workflows. Within 3 sprints, the team started resolving blockers independently without escalation.”

🔟 How do you apply servant leadership in a complex enterprise?

Approach:

- Remove **impediments** and **protect teams from unnecessary pressure**.

- **Enable decision-making** by providing information, training, and coaching.
- Advocate for **continuous improvement, collaboration, and transparency**.
- Balance **stakeholder expectations** with team needs.

Example:

> "In a multi-ART insurance program, I facilitated cross-team coordination, resolved dependencies, and shielded the teams from last-minute executive demands. This allowed teams to deliver high-value increments predictably."

✅ Quick Summary Table

Question	**Key Actions**	**Example Impact**
Resolve conflict	Listen, facilitate structured discussion, find solution	Cleared blockers quickly
Coach senior management	Workshops, metrics, show business value	Execs supported Agile adoption
Micromanaging PO	Educate, maintain DoD, promote autonomy	Team regained focus
Waterfall → Agile	Training, incremental change, quick wins	Teams delivered working increments
Retrospectives	Structured formats, actionable items	QA cycle reduced 20%
Agile maturity	Metrics & surveys	Improved flow & predictability
Build trust	Transparency, delivery, feedback	Reduced escalations
Missing PI objectives	Root cause analysis, I&A, adjust scope	Next PI 90% objectives met
Self-organization	Encourage ownership, gradual coaching	Teams resolved blockers independently
Servant leadership	Remove impediments, shield, enable decisions	High-value delivery, predictable outcomes

👋 -- SAFe Scaling & Program Management -- 👇

1️⃣ How do you handle dependencies across multiple ARTs?

Approach:

- Identify dependencies during **PI Planning** and pre-planning sessions.
- Use **Program Boards / Solution Boards** to visualize dependencies across ARTs.
- Conduct **Scrum of Scrums and PO Syncs** across ARTs for coordination.
- Assign clear owners for each dependency and track progress in tools like Jira Align.
- Encourage **decoupling architecture** where feasible to reduce cross-ART blockers.

Example:

> "PolicyCenter and BillingCenter ARTs had overlapping features. We visualized dependencies on the program board and scheduled weekly cross-ART syncs, reducing blockers by 40%."

2️⃣ How do you measure PI performance and success?

Metrics:

- **PI Predictability:** % of PI Objectives completed vs committed.
- **Feature Completion:** Number of features delivered vs planned.
- **Value Delivery:** Business value achieved (weighted by priority).
- **Quality / NFR Compliance:** Defect density, automated test coverage.
- **Team Engagement / Improvement Actions:** Completion of retrospective action items.

Example:

> "We measured PI performance by tracking objectives completion and NFR compliance. After implementing dependency management improvements, predictability increased from 65% to 90% in the next PI."

3️⃣ How do you facilitate risk management in SAFe?

Approach:

- Use **ROAM (Resolved, Owned, Accepted, Mitigated) framework** during PI Planning.

- Maintain **risk registers** linked to features or objectives.
- Review and update risks in **Scrum of Scrums or ART syncs**.
- Encourage early discussion of **technical, business, and external risks**.

Example:

> "We identified potential cloud service delays during PI Planning. Assigned mitigation ownership and tracked progress in Jira Align, preventing schedule slippage."

4️⃣ What is the difference between PI Objectives, Features, and Epics?

Term	Definition	Example
Epic	Large initiative spanning multiple PIs	"Implement full self-service portal for policyholders"
Feature	Mid-level capability delivering specific business value within PI	"Online policy renewal functionality"
PI Objective	Team-specific commitment during a PI	"Automate policy renewal notifications for X product line"

5️⃣ How do you align team backlogs with program-level objectives?

Approach:

- Map **team stories and tasks** to program-level **features and PI objectives**.
- Use **backlog refinement sessions** to ensure alignment.
- Track progress with **Jira Align, Azure DevOps, or Confluence dashboards**.
- Facilitate **PO Syncs** and program board reviews for transparency.

Example:

> "We linked team backlog items to program features in Jira Align. During PI Planning, teams ensured all high-priority objectives were covered, resulting in 95% alignment."

6️⃣ How do you deal with distributed ARTs across multiple geographies?

Approach:

- Schedule **overlapping hours** for ceremonies.
- Use **collaboration tools** (Zoom, Miro, Jira, Confluence) for transparency.
- Record demos and meetings for asynchronous viewing.
- Foster **team culture** via virtual workshops, recognition, and rotation of meeting times to balance fairness.

Example:

> "Managed ARTs across US, Europe, and India. Using Miro for dependency mapping and virtual PI Planning, engagement improved, and release predictability increased."

7️⃣ How do you handle inter-team conflicts at the program level?

Approach:

- Identify **root cause**: resource contention, priority conflicts, or misaligned objectives.
- Facilitate **structured discussions** with teams and POs.
- Focus on **program goals, not individual preferences**.
- Escalate to RTE or PMO only if needed.

Example:

> "Two ARTs competed for shared resources. I facilitated a prioritization workshop using business value scoring, resolving conflicts without impacting delivery."

8️⃣ How do you track metrics like Predictability, NFR compliance, and Feature completion in SAFe?

Metrics & Tools:

- **Predictability:** Compare committed PI objectives vs achieved.
- **NFR Compliance:** Track via automated testing, static code analysis, or Jira QA dashboards.
- **Feature Completion:** Count of features completed vs planned; tie to business value.
- Visualize trends via **Program Board, Jira Align dashboards, or Tableau reports**.

Example:

> "We tracked NFR compliance through SonarQube metrics, predictability via Jira Align PI objectives, and feature completion in burn-up charts — leading to more accurate forecasting."

9️⃣ How do you coach leadership on Lean Portfolio Management?

Approach:

- Educate on **portfolio vision, value streams, and budgeting**.
- Use **metrics to show impact of prioritization and flow** (e.g., cost of delay, ROI).
- Promote **lean governance** — decision-making closer to value streams.
- Encourage **continuous funding and incremental delivery** over large projects.

Example:

> "I coached leadership to adopt Lean Portfolio Management by visualizing business value delivery per epic. This led to faster funding approval for high-priority initiatives."

🔟 How do you implement continuous improvement at program and portfolio levels?

Approach:

- Conduct **Inspect & Adapt workshops** at PI end.
- Track **action items from retrospectives** and program-level metrics.
- Promote **metrics-driven decision-making** and **root cause analysis** for systemic issues.
- Encourage **feedback loops across ARTs and portfolios**.

Example:

> "After multiple PIs, we identified recurring defects due to incomplete test coverage. We introduced cross-ART automated testing standards and reduced post-release defects by 30%."

✅ Quick Reference Table

Question	Key Actions	Example Impact
Cross-ART dependencies	Program Board, SoS, assign owners	Reduced blockers 40%

PI performance	Predictability, feature completion, NFR compliance	Predictability 90%
Risk management	ROAM, registers, ART sync	Prevented cloud delays
PI Obj vs Feature vs Epic	Team vs Program vs Large initiative	Clear mapping & planning
Align backlogs	Map stories to features	95% alignment
Distributed ARTs	Overlaps, tools, async, culture	Improved engagement
Inter-team conflicts	Root cause, workshop, prioritize	Resolved resource contention
Track metrics	Dashboards, Jira Align, SonarQube	Accurate forecasting
Coach leadership	LPM workshops, ROI, decision-making	Faster funding approval
Continuous improvement	I&A, metrics, feedback loops	30% reduction in defects

👏 --Metrics, Delivery & Leadership Challenges-- 👇

1️⃣ How do you report Agile metrics to executives?

Approach:

- Focus on **business outcomes, not just process metrics**.
- Highlight **predictability, value delivered, feature completion, and risk status**.
- Use **dashboards and visualizations** (burn-up charts, PI progress, KPI metrics).
- Provide **context with trade-offs** — e.g., scope vs quality vs speed.

Example:

> "For executives, I present PI burn-up charts, predictability metrics, and business value delivered. This shows how delivery aligns with strategic goals and supports decision-making."

2️⃣ How do you maintain quality while accelerating delivery?

Approach:

- Implement **Definition of Done (DoD) and non-functional requirements (NFR) compliance**.
- Promote **automated testing, CI/CD pipelines, and peer reviews**.
- Foster a **shift-left mindset**, integrating QA early in development.
- Monitor **defect trends and velocity** to ensure quality is not compromised.

Example:

> "In a Guidewire implementation, we accelerated delivery by parallelizing work but enforced automated regression tests and DoD adherence, reducing production defects by 25%."

3️⃣ How do you handle scope creep in large programs?

Approach:

- Use **change control via Program Backlog and PI Planning**.
- Evaluate **impact on objectives, dependencies, and delivery timeline**.
- Negotiate **trade-offs with stakeholders**, prioritizing based on business value.
- Document changes and **communicate implications** to the ARTs.

Example:

> "During a PI, a regulatory change was requested. We reprioritized the backlog, adjusted features, and communicated the trade-off to stakeholders, ensuring delivery of critical objectives without overloading teams."

4️⃣ How do you manage remote or hybrid SAFe teams?

Approach:

- Schedule **overlapping hours for ceremonies**.
- Use **collaboration and visualization tools** (Miro, Confluence, Jira Align).
- Encourage **asynchronous updates** for non-critical discussions.
- Maintain **team culture** via virtual workshops, recognition, and interactive retrospectives.

Example:

> “For distributed ARTs across US, Europe, and India, we ran virtual PI Planning with Miro boards, recorded demos for asynchronous viewing, and fostered engagement with team recognition programs.”

5️⃣ How do you foster innovation and Lean thinking in a conservative organization?

Approach:

- Start with **small experiments / POCs** to demonstrate value.
- Encourage **feedback loops and iterative improvements**.
- Use **data-driven results** to show benefits of Lean thinking.
- Promote **communities of practice or innovation guilds**.

Example:

> “We introduced a microservice POC for claims processing. By demonstrating faster cycle times and reduced errors, leadership approved wider Lean adoption.”

6️⃣ How do you balance delivery pressure and team well-being?

Approach:

- Monitor **team workload and capacity** to prevent burnout.
- Encourage **healthy WIP limits and realistic PI commitments**.
- Promote **psychological safety and open communication**.
- Recognize team achievements to **boost morale**.

Example:

> “During a high-pressure release, I implemented rotating pair programming and short ‘well-being’ check-ins. Teams maintained productivity while stress levels decreased.”

7️⃣ How do you implement continuous improvement across multiple teams?

Approach:

- Conduct **Inspect & Adapt (I&A) workshops** at the ART level.
- Track **metrics like velocity, NFR compliance, defect leakage** across teams.

- Encourage **cross-team retrospectives and knowledge sharing**.
- Implement **improvement backlog items** visible to all ARTs.

Example:

> "We introduced cross-ART retros and shared improvement backlog. Automated testing standards became common practice, reducing defects across 6 teams by 30% in the next PI."

8️⃣ How do you coach multiple Scrum Masters?

Approach:

- Provide **regular coaching sessions** on SAFe principles, facilitation, and servant leadership.
- Conduct **peer learning sessions and communities of practice**.
- Support Scrum Masters in **resolving escalated impediments** and handling cross-team dependencies.
- Encourage metrics-driven improvement and consistent practices across ARTs.

Example:

> "I coached 5 Scrum Masters in a large insurance program. After introducing joint problem-solving sessions and consistent metrics reporting, team predictability improved 20%."

9️⃣ How do you handle changes to PI mid-cycle?

Approach:

- Evaluate **criticality and impact** of the change.
- Update **program backlog and re-align PI objectives**.
- Communicate **trade-offs to stakeholders**.
- Adjust team scope **without overloading**.

Example:

> "Mid-PI, a compliance requirement emerged. We added it to the backlog, deferred a lower-priority feature, and communicated updated objectives to all teams, completing critical work without delay."

🔟 How do you recover from a failed PI or ART?

Approach:

- Conduct a **root-cause analysis** during Inspect & Adapt.
- Identify **systemic blockers, dependencies, or process gaps**.
- Re-plan the next PI with **realistic objectives and backlog refinement**.
- Implement **continuous improvement actions** to prevent recurrence.
- Communicate lessons learned to leadership for transparency.

Example:

> "A PI failed due to unresolved dependencies and overcommitment. We mapped dependencies visually, adjusted objectives, and improved cross-ART coordination. The next PI achieved 95% of objectives successfully."

✅ Quick Reference Table

Question	Key Actions	Example Impact
Report metrics	Dashboards, business value, context	Execs see strategic outcomes
Maintain quality	DoD, NFR, automation, shift-left	25% fewer defects
Scope creep	Backlog reprioritization, trade-offs	Critical objectives delivered
Remote/hybrid teams	Overlaps, tools, async, culture	Improved engagement
Foster innovation	POCs, feedback loops, guilds	Leadership approved Lean adoption
Balance pressure	Monitor capacity, WIP limits, recognition	Productivity maintained, stress reduced
Continuous improvement	I&A, cross-team retros, metrics	30% defect reduction
Coach Scrum Masters	Workshops, peer learning, guidance	20% predictability improvement
Mid-PI changes	Evaluate, re-align, communicate	Critical compliance delivered
Recover failed PI	Root cause, realistic planning, improvements	Next PI 95% objectives met

🌀 Ultimate Scrum Master / SAFe Prep Sheet

1️⃣ Scrum & Agile Fundamentals

Question	Answer / Example
What is Scrum?	Agile framework for delivering value in iterations (Sprints) using roles, events, and artifacts. Example: Deliver insurance features in 2-week sprints.
3 Pillars of Scrum	Transparency, Inspection, Adaptation. Example: Daily stand-ups + retrospectives ensure visibility and improvement.
5 Scrum Values	Commitment, Courage, Focus, Openness, Respect. Example: Team commits to sprint goal and communicates blockers openly.
Scrum Roles	Product Owner, Scrum Master, Development Team. Example: PO prioritizes backlog; Scrum Master removes impediments.
Scrum Artifacts	Product Backlog, Sprint Backlog, Increment. Example: Increment = working claims calculation feature.
Scrum Events	Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective.
Timebox for Sprint	Typically 2–4 weeks. Example: 2-week sprint delivering policy renewal functionality.
Who owns Product Backlog?	Product Owner.
Who removes impediments?	Scrum Master.
Cross-functional team	Team with all skills needed to deliver a product increment. Example: Dev + QA + BA working on a claims feature.

2️⃣ Scrum Events & PI Concepts

Question	Answer / Example
Sprint Planning Purpose	Plan what will be delivered in the sprint and how.
Output of Sprint Planning	Sprint Backlog, defined sprint goal.
Daily Scrum	15-min sync on progress, plan, and impediments.
Who attends Daily Scrum?	Development team (Scrum Master facilitates; PO optional).
Sprint Review	Inspect increment, gather stakeholder feedback. Example: Demo new self-service portal features.
Sprint Retrospective	Discuss process improvements and team learnings.
Scrum Master in Retrospective	Facilitate, encourage participation, ensure actionable outcomes.
Product Increment	Working, tested, shippable functionality.
DoD	Criteria to declare a story done. Example: Code complete, unit-tested, integrated, and documented.
DoR	Criteria for story readiness before sprint. Example: Well-defined, estimated, dependencies known.

3️⃣ SAFe Fundamentals

Question	Answer / Example
SAFe	Scaled Agile Framework for enterprise-level Agile delivery.
SAFe Levels	Team, Program, Large Solution, Portfolio.
Key Roles	RTE, Product Management, System Architect, Scrum Masters, Product Owners.
Program Increment (PI)	Timebox (~8–12 weeks) of value delivery across ARTs.
Agile Release Train (ART)	Long-lived team of Agile teams aligned to a value stream.

PI Planning	Face-to-face or virtual event to plan PI objectives and dependencies.
System Demo	Review integrated increment of all teams' work.
Feature vs Epic	Feature = mid-level capability; Epic = large initiative spanning multiple PIs.
Team PI Objectives vs Program PI Objectives	Team = deliverables for sprint/PI; Program = high-level outcomes for ART.

4️⃣ Metrics & Reporting

Question	Answer / Example
Track Agile Metrics	Velocity, predictability, NFR compliance, PI objective completion, defect trends.
Burndown/Burn-up charts	Track remaining work vs progress; burn-up shows scope change.
Retrospective outcomes	Track improvement actions in backlog; review next sprint/PI.
Technical debt	Track known debt; create backlog items to address.
Velocity trends	Predict delivery and capacity planning.
Report to Executives	Focus on business value, predictability, feature completion; use dashboards.

5️⃣ Scrum Master / ART Coaching

Question	Answer / Example
Handle PO changing priorities	Educate on sprint goals, maintain backlog transparency, negotiate trade-offs.
Ensure DoD	Coach teams on DoD, integrate into CI/CD and QA processes.
Handle unplanned work	Evaluate urgency; defer or swap stories to maintain sprint goal.

Motivate demotivated team	Recognize achievements, remove blockers, promote psychological safety.
Coach new PO	Guide on backlog prioritization, stakeholder alignment, and SAFe principles.
Coach multiple Scrum Masters	Conduct workshops, peer learning, standardize metrics.
Coach leadership	Lean Portfolio Management, value-driven funding, Agile principles with metrics.
Apply servant leadership	Remove impediments, shield teams, enable decision-making.
Self-organization	Encourage ownership, WIP limits, peer accountability.

---

6 Team & Program Challenges

Question	**Answer / Example**
Handle cross-team collaboration	Program Board, Scrum of Scrums, joint refinement sessions.
Handle distributed ARTs	Overlapping hours, collaboration tools, async communication, foster culture.
Handle conflicts	Facilitate discussion, focus on goals, escalate only if needed.
Scope creep	Evaluate impact, negotiate trade-offs, communicate to stakeholders.
Remote/hybrid teams	Tools, recordings, virtual workshops, overlapping time zones.
Innovation in conservative org	POCs, small experiments, demonstrate value, guilds/CoPs.
Balance delivery & well-being	Monitor capacity, WIP limits, recognition, rotation, check-ins.
Handle team missing PI objectives	Root cause analysis, adjust scope, improve dependency management.
Mid-PI changes	Evaluate criticality, re-align backlog, communicate trade-offs.

Recover failed PI/ART	Root cause, realistic planning, continuous improvement actions.

7️⃣ SAFe Alignment & Continuous Improvement

Question	Answer / Example
Align backlogs to program objectives	Map stories to features, use PI planning, track in Jira Align.
Track metrics: Predictability, NFR, Feature Completion	Dashboards, burn-up charts, automated testing tools.
Facilitate PI Planning	Pre-plan dependencies, ART syncs, business context alignment.
Facilitate risk management	ROAM, risk register, ART sync, mitigation ownership.
Implement continuous improvement	I&A workshops, cross-team retros, improvement backlog, metrics.
Foster Lean thinking	Data-driven results, incremental delivery, communities of practice.

Tips for Interview Use:

1. **Structure answers**: Situation → Action → Result (SAR).
2. **Use real examples**: Tie answers to P&C Insurance or enterprise experience.
3. **Emphasize leadership & coaching**: Especially servant leadership, Lean Portfolio Management, and cross-team facilitation.
4. **Metrics matter**: Executives care about predictability, ROI, NFR compliance, and value delivery.

🎯 Agile & SAFe Quick Visual Cheat Sheet (1 Page)

1️⃣ Scrum Roles

Role	Responsibility
Product Owner (PO)	Defines & prioritizes backlog, ensures business value.
Scrum Master (SM)	Removes impediments, facilitates ceremonies, coaches team.
Development Team	Self-organizing, delivers increments, cross-functional.

2️⃣ Scrum Events

Event	Timebox	Purpose
Sprint Planning	2–4 hrs / week of sprint	Plan sprint backlog & goal.
Daily Scrum	15 mins	Sync progress, plan, impediments.
Sprint Review	1–2 hrs	Demo increment, get stakeholder feedback.
Sprint Retrospective	1–2 hrs	Inspect & improve process.

3️⃣ Scrum Artifacts

- **Product Backlog** → Ordered list of features & stories.
- **Sprint Backlog** → Selected stories for sprint.
- **Increment** → Shippable, tested work.
- **DoD / DoR** → Criteria for ready/done.

4️⃣ SAFe Hierarchy

`Portfolio → Large Solution → Program (ART) → Team`

- **Epic** → Large initiative spanning multiple PIs
- **Feature** → Mid-level capability delivered by ART
- **User Story** → Team-level work item

- **PI Objectives** → Team commitments for PI
- **Program Increment (PI)** → 8–12 weeks of value delivery
- **Agile Release Train (ART)** → 5–12 Agile teams delivering value

---

5️⃣ Key SAFe Ceremonies

Ceremony	Purpose
PI Planning	Align ARTs, plan PI objectives, identify dependencies
Scrum of Scrums	Cross-team coordination, track dependencies & risks
System Demo	Show integrated increment across ARTs
Inspect & Adapt (I&A)	Identify improvements, root cause analysis, plan next PI

6️⃣ Key Metrics (Team & Program)

Metric	Purpose
Velocity	Predict delivery capability
Capacity	Max work team can handle per sprint
Predictability	% PI Objectives achieved vs committed
Feature Completion	# Features delivered vs planned
NFR Compliance	Non-functional requirement adherence
Burn-down / Burn-up	Track progress & scope changes
Technical Debt	Track and manage backlog items for quality

7️⃣ Cross-Team Coordination

- **Dependencies** → Visualize on Program Board, assign owners.
- **Risks** → ROAM framework (Resolved, Owned, Accepted, Mitigated).
- **Backlog Alignment** → Map team stories to program features and PI objectives.
- **Distributed Teams** → Overlap hours, async tools, recorded demos.

8️⃣ Servant Leadership & Coaching

- Remove impediments & shield teams
- Encourage self-organization & ownership
- Coach PO & leadership on Agile / Lean Portfolio Management
- Foster psychological safety, motivation, and continuous improvement

---

9️⃣ Quick Reference Tips

- **Scope Creep** → Evaluate impact, negotiate trade-offs, communicate.
- **Failed PI / ART** → Root cause analysis, realistic re-planning, continuous improvement.
- **Innovation in conservative org** → Start with POCs, show incremental value.
- **Balance delivery & well-being** → WIP limits, check-ins, recognition.

🎯 Additional Topics

🧩 1. Scrum Framework – Missing Deep Details

Already covered: Roles, artifacts, events, DoR/DoD
Add these:
✅ *Sprint Goal* — The single objective that gives focus to the Sprint.
✅ *Increment Review Criteria* — How to evaluate business value at the end of Sprint.
✅ *Refinement / Grooming Sessions* — Continuous backlog refinement (~10% of team capacity).
✅ *Definition of Ready (DoR)* vs *Definition of Done (DoD)* with examples.
✅ *Release Planning* — How multiple Sprints align toward a release.

Example to remember:

> "Sprint Goal = Purpose; Increment = Output; DoD = Quality Gate."

🚀 2. SAFe Framework – Missing Enterprise-Level Concepts

Already covered: PI, ART, Ceremonies, Roles, Metrics
Add these:
✅ *Lean-Agile Principles* — The 10 foundational SAFe principles (economic view, decentralization, cadence/synchronization, etc.)
✅ *Core Competencies of SAFe* — 7 competencies like Agile Product Delivery, Team & Technical Agility, Lean Portfolio Management, etc.
✅ *Value Stream* — End-to-end flow of value from concept to customer.
✅ *Enablers* — Technical tasks or spikes that support features/architecture.
✅ *Architectural Runway* — Existing technical infrastructure to support future development.
✅ *Portfolio Kanban* — Flow of Epics through Portfolio-level governance.
✅ *Solution Train* — Coordination layer for very large systems (multiple ARTs).

📊 3. Metrics & Reporting – Add Outcome-Focused Metrics

Already covered: Velocity, Predictability, NFR, Feature Completion
Add these:
✅ *Cycle Time* — Time from work start to finish.
✅ *Lead Time* — Time from request to delivery.
✅ *Escaped Defects* — Defects found post-release.
✅ *Team Happiness Index / Engagement Score* — For psychological safety & morale.
✅ *Flow Efficiency* — Ratio of active work to total elapsed time.
✅ *Cumulative Flow Diagram (CFD)* — To visualize WIP & bottlenecks.

🧠 4. Leadership & Coaching Mindset

Already covered: Servant leadership, coaching teams
Add these:
✅ *Coaching Stance* — When to mentor, facilitate, coach, or teach.
✅ *Conflict Management Styles* — Competing, Collaborating, Avoiding, Accommodating, Compromising.
✅ *Emotional Intelligence (EQ)* — Awareness, empathy, and influence in Agile teams.
✅ *Agile Maturity Models* — Levels of adoption and how to measure progress.

🛠️ 5. Tools & Reporting Dashboards

Add these practical tools (often asked in interviews):
✅ *JIRA / Azure DevOps (ADO)* — Sprint boards, burndown, velocity tracking.
✅ *Confluence / Miro / Mural* — For retrospectives, PI planning, and visualization.
✅ *Power BI / Tableau* — For program-level Agile reporting and OKRs.

🧩 6. Advanced SAFe Roles (for Program & Portfolio levels)

✅ *Release Train Engineer (RTE)* — Facilitates ART processes & PI execution.
✅ *System Architect / Engineer* — Defines technical vision and architectural runway.
✅ *Business Owner* — Ensures business outcomes align with PI objectives.
✅ *Lean Portfolio Management (LPM)* — Aligns strategy, investment, and execution.

🧭 7. Continuous Improvement Frameworks

✅ *Inspect & Adapt (I&A)* workshop in detail (metrics review + problem-solving).
✅ *DevOps in SAFe* — CALMR approach (Culture, Automation, Lean Flow, Measurement, Recovery).
✅ *Continuous Exploration, Integration, and Deployment (CE, CI, CD)* — How value flows end-to-end.

✅ **In summary**, here's the **full readiness coverage** (✓ = done, ➕ = suggested to add):

Category	**Coverage**
Scrum Roles, Events, Artifacts	✓
DoR / DoD / Sprint Goal	➕
SAFe Levels, ART, PI	✓
Value Stream, Enablers, Runway	➕
Metrics (Velocity, Predictability)	✓
Flow Metrics (Lead/Cycle Time, CFD)	➕
Leadership & Coaching	✓
Conflict & Maturity Models	➕
Agile Tools (JIRA, ADO)	➕
Program & Portfolio Roles	➕
Continuous Improvement (DevOps, I&A)	➕