Task 4 Part 2

**A. Challenge from Table 8 and Humble/Virmani**

One challenge listed in Table 8 is **lack of awareness and transparency among team members**. This means that sometimes people in the team don’t really know what’s happening in the project , like what features are being worked on, what’s been deployed, or if there are any problems in the pipeline.

Jez Humble, in his book, talks about how important it is to have **fast feedback** and **visibility** in the software delivery process. He recommends using things like dashboards, automated builds, and deployment pipelines so everyone can see the current status of the code and systems. This helps prevent surprises and lets teams respond to issues quickly.

Virmani also agrees this is an issue. He talks about DevOps being all about **collaboration** between teams, especially between developers and operations. He focuses more on building a **culture of teamwork** and making sure people share knowledge and responsibility.

So , both Humble and Virmani agree that this is a challenge, and they both suggest similar solutions: better communication, shared tools, and working closely as one team instead of separate groups.

**B. Challenge similar to Agile or Lean**

One of the challenges listed in Table 8 is **coordination and collaboration between teams**, and this is something that’s also very common in Agile or Lean software development.

In Agile, we already focus a lot on teamwork. For example, teams have daily standups where they talk about what they’re working on, sprint planning to organize tasks together, and retrospectives to reflect and improve. These practices are all about improving how team members communicate and work together.

In DevOps, this challenge becomes even bigger. It’s not just about developers and testers anymore. DevOps also brings in **operations teams** the people who handle deployments, monitor systems, manage servers, and keep the software running after it’s released.

So now, instead of just coordinating within one development team, we have to **connect and collaborate across multiple teams** with different roles and responsibilities. That can be hard, especially if those teams are used to working separately.

But in a way, this DevOps challenge is just a **natural extension** of what Agile is trying to do which is to bring people together, break down silos, and deliver value faster. DevOps simply continues that same goal, but now across the entire software delivery pipeline, all the way to production.

**C. Practice to start with (from Table 9)**

If I was leading a company that’s moving from traditional (Waterfall-style) software development to DevOps, I would start with the practice: **“Improve Team Awareness and Communication.”**

This is important because DevOps is not just about tools ,it's about people working together more closely. In many traditional teams, developers, testers, and operations people often work in silos, meaning they don't talk much or share updates. That leads to misunderstandings, delays, and mistakes, especially during releases.

By improving awareness and communication, everyone knows what’s happening in the project, what’s being deployed, and what changes might affect them. It builds trust and reduces fear around automation and faster releases.

Some simple things we could start with include:

* **Changelogs** that show what code was changed
* **Version tags** in the repository
* **Slack or email alerts** when builds succeed or fail
* **Short daily sync-ups** between dev and ops teams

Starting with this practice also helps the whole team get used to working together , which makes it easier to introduce the next steps like CI/CD tools, automated tests, or infrastructure changes.

In short, better communication helps reduce resistance, builds a DevOps mindset, and creates a smoother path for all the other changes we want to make.