WHAT CONTEXT TO PROVIDE TO FACILITATE GROWTH AREAS FOR TEAMS?

Consideration for Growing Cultural Aspects

- **Continuous Learning and Continuous Improvement**:
  - Develop multidisciplinary teams than resource pools to create shorter feedback loops to enable everyone to understand the implications of an action.
  - Conduct well-funded knowledge sessions (partly) during work time and encourage everyone that investing in these sessions is necessary. Do not forget good food and fun to make these sessions exciting to attend.
  - Conduct hackathons/innovation days to let everyone understand the importance of innovation to the business. Again, it requires encouraging everyone to invest time.
  - Build a sense of responsibility to enable people to do the expected job.
  - Advocate the concepts of a Minimal Viable Product (MVP) and encourage people to apply these when trying out new ideas.
- **Experimentation and Risk-Taking**:
  - Provide the required time and means to enable people to explore new ideas.
  - Ensure that obtaining infrastructure/playground/sandbox does not pose a hurdle to try out new ideas. Systems need to be made instantly available as soon as needed, just like in the cloud.
  - Trust people and appreciate new ideas. Provide them little space to try out new ideas and do the right job.
  - Conduct knowledge sharing sessions and offer a platform to people to showcase their new ideas.
  - Create an environment that encourages safe-to-fail experiments.
- **Build Quality in**:
  - Encourage people to be end-to-end responsible for the product or service.
  - Ask the team to provide insight into the amount of functionality that can be delivered as they know best about software development.
  - Develop transparency through feedback loops. Equip people with the means to measure their performance. Install open monitors on the work floor that provide a detailed view of the status of selected jobs and display test results in the open floor.
  - Test and deliver often. Correlating test results of tasks performed over the last couple of hours is easier compared to weeks.
  - Add to the Definition of Done (DoD), as per Scrum terminology, when delivering software. Ensure to verify the software at various stages of testing.
  - Make continuous improvement, the way of working. Ensure to reserve 10% of the time for continuous improvement. In Scrum, retrospectives focus on continuous improvement.
  - Automate manual tasks to ensure producing consistent results. Automate tests, deployments, and provisioning.

- **An Engineering Culture**:
  - Ensure matching interviewee ambitions with that of the company.
  - Assign products to people considering their “added skill or knowledge.” Avoid following the function-title model.
  - Reserve 10% of the time for automating slow, repetitive, and costly manual tasks to avoid human errors.
  - Develop a supportive culture where people help one another to solve issues other than managing utilization.
  - Provide a standard playground/sandbox to people for experimentation.
- **A Culture of Effectiveness**:
  - Begin your work considering the end in mind. Focus on doing value-add activities, ignoring the non-value-add process steps.
  - Help people build quality by not cutting corners but by automating tests and manual (often boring) activities.
  - Focus on team autonomy (independent teams) and remove the source of inefficiency by eliminating handover moments as much as possible.
  - Avoid following rigid processes without any profound thought and discussions. Allow the team to choose activities to achieve results in different situations.
  - Encourage working in small batches to assess whether everyone is heading in the right direction. Iterate when working on products.
  - Make people aware of the concept of value-add activities by providing Lean training or introducing Lean principles to enable them to start thinking about it considering their work activities.
- **A Culture of Product Thinking**:
  - Allow the entire team to connect with the customer, not just the Product owner. Ensure to provide product demo to the customer after each sprint.
  - Reduce the feedback loops by incorporating customer or user feedback straight into the user storyboard.
  - Encourage customers to provide feedback about the product and allow the team to respond to the feedback. Ensure the team understands the what and why behind the given work.
  - Make teams product-based, end-to-end responsible, and capable of handling multiple projects.
  - Aim at complete autonomy of the teams and the products.
  - Encourage teams to write about their products, such as the company blog or speak about the product at seminars/customer engagements.
- **A Culture of Taking Responsibility**:
  - Allow or support people to think of ways to perform their tasks rather than how to do it.
  - Provide the required authority to enable people to make relevant decisions as and when required. For example, let the teams decide when to go live and allow teams to access systems for which they request somewhere else.

  - Make people end-to-end responsible for a product.
  - Allow people to talk about anyone who is avoiding responsibility by providing a platform to bring transparency, such as Scrum Board, and address derailment of the process.
  - Develop a fail-to-safe environment by rewarding people for taking responsibility, a failure for experimenting out new ideas, and speaking about something that does not seem to work.
- **Inspirational and Fun Environment**:
  - Provide good coffee to the people. They drink it the most during the day.
  - Develop a friendly and open environment.
  - Bring some excitement to the workplace by investing in fun activities, such as games and an outing. These activities allow people to know each other outside their work environment.
  - Encourage the team to develop their own space to help them create a sense of ownership.