Table of Contents

1 Background

Today, more and more customers have started using agile and they are in various levels of maturity with respect to its usage, governance and metrics. Many RFPs ask for our comments with respect to customer's current ways of working. Scrum has been the de facto model used in many places. But Spotify is also prevalent and many customers have adopted a hybrid approach too.

Scrum is one of the many frameworks emerging from agile. This agile methodology focuses on being adaptive to change and creating software iteratively. It has been widely and traditionally practiced by teams of small sizes. For aligning and managing multiple teams for large or complex projects, you need an agile scaling model.

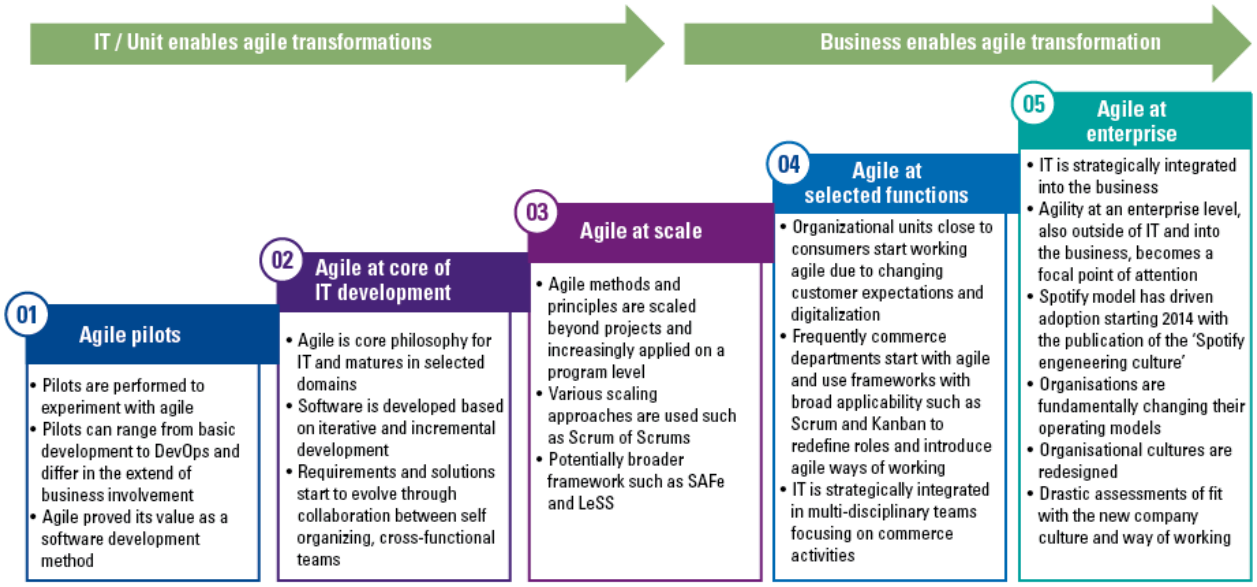
These agile scaling models provide a framework that sets down the guidelines, techniques, and workflows which ensure that working with hundreds or even thousands of practitioners, remains coordinated and easy to manage.

Spotify Agile model (a.k.a Spotify Engineering culture) can be proposed for customers looking to adopt an agile delivery model that

- can be scaled to be adopted by large programs and projects across the whole IT organization
- provides agility and faster time to market by way of creating autonomous teams having minimum dependencies and handoffs while providing at the same time alignment to organization objectives
- Creates highly skilled cross functional teams which can take idea from concept to launch
- Enables start up culture at the team level while still within the bounds of large organization structure
- Fosters innovation.

Many banks specifically, these days are tying up with Fin Tech to meet their specific technology /product needs and get benefits of fintech companies which bring specialized skills and agility. Adopting Spotify agile model can help them create a fintech kind of environment within their own organization.

A typical agile maturity / adoption road map can be as follows:



In this document, we discuss the applicability of scaled models, selection of models, a typical target operating model and finally the factors that need to be reflected in the solution and commercials.

2 SAFe

2.1 Brief Description

Scaled Agile Framework (SAFe) allows enterprises to accomplish their organizational goals to produce the highest quality product in the shortest sustainable amount of time. It is an approach that scales Scrum to an Enterprise Level and gives you the freedom to scale according to your business needs. It introduces a philosophy of servant and lean-agile leadership and goes beyond just implementing an organizational structure, rather instills a new mindset.

SAFe can be easily extended and scaled to hundreds or even thousands of team members.

2.2 Roles

Agile teams will consist of 5-12 members who are cross-functional and self-managing in nature. There is a dev team, product owner and a scrum master in every team.

- Product owner – Prepares the team backlog by prioritizing the features.
- Scrum master – Servant leaders for their agile teams as their role is extremely vital in discussing and clearing out the problems that the teams are facing. They have to be empathetic to the members of the team thus to create a healthy and productive environment for the teams to work.
- Product Manager – At a higher level than the product owner, product managers prioritize the features of the Program Backlog.
- Release Train Engineer – The ultimate leader of the Agile Release Train.
- Solution Engineer – Similar to the duties of a Release Train Engineer but leads more than one Agile Release Train.
- Enterprise Architect – Lays down the foundation of the program portfolio. They guide the direction towards creating a strategic, technical and an adaptable design.
- Epic Owner – Creates an Epic which possesses economic and business value. This epic can be any feature or a user story with requirements covering a large scope. The epic owner works directly with the Agile Release Train once their epic is selected and verified by the Lean Portfolio Management.
- Business Owner – A key stakeholder in the Agile Release Train.
- Lean-Agile Leaders – Servant leaders that teach, coach and educate the teams on how to work in SAFe.
- Lean Portfolio Management – The highest level of decision making for the strategies, portfolio operations and governance in SAFe.

2.3 Planning and Events

A major element of the Scaled Agile Framework revolves around the Program Increment (PI). Effectively planning and executing the PI is a major feat.

It is a collection of sprints that last for 8-12 weeks with sub-sprints that can last for 2-4 weeks. The date for planning the Program Increment is fixed. This ensures that it happens at a fixed cadence at a time.

It is scheduled in advance and its date is conveyed to all the members that are part of the PI. This helps to reduce costs on travel and logistics, assuring that everyone will make an attendance. Below are the five steps that describe the process of PI planning in SAFe.

Process Step	Description
Agile Release Train	• Agile teams that are specialized for working on one particular element, get together and board the Agile Release Train. These teams will essentially drive the Agile Release Train. There are 5-12 teams in the train which are geared towards delivering value and in fact build what will be planned. • The Release Train Engineer leads the train and ensures that these teams are synchronized and are in constant collaboration with each other.
Planning the PI (Program Increment)	• The planning lasts for two days typically. All the PI objectives are planned, dependencies are sorted and the teams are aligned in a cadence. • Product Management works with the product owners to prioritize and select the features which define the scope of the project. After setting the PI objectives, each Agile team works on their own set target and manages their work in independent Sprints. • Development is incremental and iterative.
Synchronization of Agile Train	• Scrum of Scrums: The Release Train Engineer acts as the chief scrum master and has the scrum of scrums where all the scrum masters of all the agile teams of the Agile Release Train are present. It is held weekly or twice a week, whatever suits them, to discuss any impediments or challenges being faced. • Product Owner Sync: All the product owners of the teams meet with the Release Train Engineer or the Product Manager to talk about how the PI objectives can be achieved.
System Demo	• Held once a week, all the features that are implemented by the Agile Release Train to present, are demonstrated to the stakeholders. • The stakeholders give their feedback. • This is beneficial as it sheds light on how well the agile teams in the Agile Release Train are integrated together. • The Agile Release Train has a better idea of what the stakeholders really want and it also gives an opportunity to improve.
Pre-Planning	• Near to the end of the PI, a pre-planning session is done to plan for the upcoming program increment. • The Release Train Engineer and the Product Managers work on refining the program backlog for mapping out the PI objectives for the upcoming Program Increment.

2.4 Levels of SAFe

The implementation of SAFe is done at four levels.

- Portfolio Level – The level at which the inception of strategies and planning of budget comes into being. It fosters the development of Value Streams thereby organizing the Lean-Agile Enterprise into delivering a solution.
- Program Level – The level which orbits around the Agile Release Train to give continuous delivery of solution to the customer. The development teams and stakeholders plan, commit,

execute, inspect and adapt in the Agile Release Train to deliver a full or part of the solution incrementally.
- Team Level – Critical for organizing and defining the roles necessary in the Agile Teams that powers the Agile Release Train. It is part of the Program Level where the Agile Release Trains develops, tests and delivers the working software at the end of an iteration. Ensures coordination with multiple Agile Teams to create an integrated system.
- Large Solution Level – An optional level in SAFe to build large solutions requiring collaboration from multiple Agile Release Trains which may have thousands of practitioners. It demands a Lean-Agile approach at a very large scale. Economic framework and financial boundaries support this level for the Large Solution Level. It is organized around the Program Increment, which is connected with the Agile Release Trains in the Large Solution Level.

2.5 Retrospectives

At the end of the PI, there is an Inspect and Adapt session. This event sums up the entire work done by the Agile Release Train. It is a staged event that is attended by the stakeholders, business owners, agile teams and customers. It is organized by the Product Management and supported by the Release Train Engineer.

The PI is evaluated by the Program Increment Performance Report, where the business owners rate the business value. This business value helps attain the achievement percentage.

Followed by the Program Predictability measure, these findings help identify any issues with the Program Increment. Once the issues are identified, Root Cause Analysis is used and then brainstorming for finding the solutions.

Any organization can adopt SAFe but it is not an overnight transition There are defined guides available on the Scaled Agile Framework's website that gives a step by step insight into transitioning to SAFe.

Key Features

- Designed for large enterprises, makes it easier to work with multiple teams.
- Well defined organizational structure, with designated roles and responsibilities.
- Lean processes that mean there is a minimal waste, thus ensuring that the teams focus on what really needs to be done.
- The formation of the Agile Release Train helps to maintain collaboration amongst teams.
- Very well documented which is available for consultation.
- Comprehensive and innovative approach as the teams relentlessly improve and adopt newer ways of working.
- Limits the Work in Process (WIP) to keep the teams more focused to produce a higher quality of work.
- Elaborate handling of processes from the team level to the high level. An active interaction between the development team and the team consisting of the Vice President and the C-Level individuals.
- Promotes trust, collaboration and transparency between the development and the top management.
- Emphasizes on attaining business value in the shortest sustainable time.
- Ensures consistent approach towards planning, execution and delivery.

- Promotes the sharing of strategy, common vision and architecture amongst the development and the managerial teams.
- Constant feedback from customers helps maintain a successful business relationship leaving room for improvement throughout the entire process.

3 LeSS (Large Scaled Scrum)

3.1 Brief Description

Large Scaled Scrum, abbreviated as LeSS, is one of the leading frameworks of agile software development. It is a multi-team scrum framework which can be applied to an agile team consisting of twelve, hundreds or even thousands of individuals, all of whom are working together on one specific shared product.

Using LeSS you can create large or small sized products. It is a simple and minimalistic framework where there is less enforcement of rules, processes, roles or artefacts. There are only conventional scrum roles such as the product owner, scrum master and the team.

LeSS is very customer-centric as teams get to interact directly with the customer while the product owner focuses on setting the roadmap, priorities and the long-term vision of the product.

There are two types of LeSS:

- Basic LeSS is for 2-8 Teams
- LeSS Huge is for more than 8 Teams.

Multiple teams can work. It can be scaled by having LESS huge which is implementing multiple basic LeSS frameworks altogether.

3.2 Roles

Teams – There are maximum 7 team members. They are cross-functional and self-managing. They are also called Feature Teams.

- Scrum Master – Facilitates 1-3 teams. A scrum master guides and teaches the teams on how to work in LeSS.
- Product owner – Manages the product backlog which consists of the list of features. One product owner can only manage up to 8 teams.
- Area product owner – When there is LeSS huge they are responsible for their respective team product backlogs. They can work with eight teams.

3.3 Planning and Events

- A deliverable product is created in every sprint. These sprints may last for 1-4 weeks. The development is iterative and incremental.

- The first stage of the sprint planning involves the selection of items from the product backlog. Two members of each team, meet with their product owner to make the selection of high priority items from the backlog.
- In the second stage of planning, the team discusses selected items. Once a team has chosen its items from the product backlog, planning is done to achieve the sprint goals.
- There is also a product backlog refinement session. The customer and the teams discuss how the existing requirements can be improved or if new requirements should be added. This session is also essential in talking about what work needs to be done in the upcoming sprints.
- DevOps and Continuous Integration is key for smooth delivery to the customer. A team should deliver a shippable increment at the end of every sprint.
- There is an Undone Department where there is a list of tasks that were not done in a sprint. Normally, this department does not exist. But sometimes the teams are not able to complete their tasks so they are shifted to the next sprint. This is the case when creating a large project while working in LeSS Huge and incorporated with feature teams.

3.4 Retrospectives

There is also a retrospective where all the teams, product owners, scrum masters and the management work to understand any impediments that affect the delivery of the product. Teams regularly have their own retrospectives, reviewing what is done to continuously improve.

In order to embrace LeSS completely, the organizational structure is completely different from traditional program management. It is recommended by LeSS to start applying principles of LeSS with one scrum team and adapt the change step by step.

3.5 Features

- LeSS provides the entire product view which guarantees transparency in the work you do.
- The teams are in direct contact with the customer which enables the teams to grasp the actual idea of what the customer really needs.
- With lean thinking, there is minimal waste, thus ensuring focus on what really needs to be done.
- There is ample room for the team to learn and grow consistently.
- Teams are feature-oriented, customer-centric and their approach is multi-component.
- Dependencies are handled at the integration level by sharing the code base with other teams. More frequent code integration is recommended to avoid complexities.
- The role of management is focused on defining the vision and nurturing of the team members. Product Owner defines and prioritizes the high-level requirements for the teams.
- Teams coordinate with each other frequently and share the code base.
- There are design and architecture workshops to align synergy across all the teams and focus towards the end product.
- Frequent retrospectives and inspect and adapt sessions are helpful in ensuring continuous improvement.
- Heavily focused on the Product Owner.
- No guidelines on portfolio management.
- Items are in basic LeSS while Epics exist in LeSS Huge.
- The role of the scrum master fades away once the teams become proficient in LeSS.

4 Nexus

4.1 Brief Description

Nexus is a simple framework which implements scrum at scale across multiple teams to deliver a single integrated product. Teams work in a common development environment and are focused on producing a combined increment every sprint with minimal dependencies.

It can be applied to 3-9 scrum teams. So, it cannot be scaled to more than 9 teams and not more than a hundred practitioners.

4.2 Roles

Teams: 5-12 members that are Self-managing and cross-functional.

- Scrum Master
- Product Owner
- Nexus Team: Constituted of 1-2 members from each scrum team. Responsible for planning the vision and the bigger picture of the overall product.
- Nexus Integration Team: Essential for keeping multiple teams technically and successfully integrated.

4.3 Planning

- The Product Owner comes up with a refined Product Backlog. The teams select items from the Nexus Product Backlog. The backlog can have stories, tasks, business initiatives, epics or any item of any size that suits the teams.
- Items in the product backlog are continually refined to minimize or clear away any dependencies. New requirements can also be added. The product owner is the ultimate responsibility for the backlog but if the size of the team exceeds then they may have to delegate some of their tasks to the scrum team, business analysts, project managers or other roles.
- Then there is a sprint planning session that has two parts.
- In the first part, the Nexus Team conducts a sprint planning session in which they plan the bigger picture of the project. Information and decisions made based on dependencies. Scrum Teams work on their individual sprint backlog. During these sprint planning session, teams interact and collaborate with each other. Teams align themselves in order to achieve their sprint goals. At the completion of all teams' sprint backlog planning, Nexus Sprint Backlog gets ready. It is a collection of sprint backlogs of each team which envisions the bigger picture.
- Scrum teams have their own scrum cadence in which they create a sprint routine and have their respective sprint backlog.

4.4 Retrospectives

A Nexus Sprint Review is held at the end of the sprint in which all the scrum teams meet with the product owner and review the integrated increment. Scrum teams do not have their own sprint reviews. There is only one collective sprint review in which the integrated increment is the subject.

Finally, there is a Nexus Sprint Retrospective. The essence of every retrospective is to meet and identify shared challenges. The solutions are discussed by sharing ideas and how they can improve. The Nexus Team and the scrum teams have their individual retrospectives. Then there is a collective retrospective where solutions are shared with the entire nexus and the scrum teams.

No change in existing organizational structure is needed. It can be adopted easily in the current organization all you need is knowledge of Scrum.

4.5 Features

- This framework promotes and ensures transparency, continuous integration and relentless improvement.
- Having a single product and sprint backlog boasts transparency as all the teams sprint data can be easily visualized. Daily scrums enhance communication and help erase dependencies.
- Working in a shared environment where work is constantly being integrated into one.
- Final product guarantees continuous integration.
- Teams use automation to manage any complexities.
- The Nexus Integration teams give the necessary support and facilitation to the teams to keep in them in line.
- Thus, eliminating the need of scrum of scrums meeting that is an essential part of other scaling frameworks. They make sure if the processes are followed and truly work as servant leaders to ensure that the teams flourish.
- The teams confirm relentless improvement with activities like the refinement of the product backlog, sprint review and retrospective.
- Listening to feedback from the stakeholders is paramount. It is vital to adapt to any changing requirements and to eliminate any waste with Lean-Thinking.

5 Spotify

5.1 Brief Description

Spotify has become a popular music player well known for providing original and a limitless collection of music content. It was launched in 2008 and has now grown manifold. They now have 30 agile teams that are spread over 4 cities in 3 different time zones. They owe their success to their deeply rooted agile methodologies and the utilization of scaling agile, with their own flavour.

Although the makers of this agile scaling method do not guarantee that it will work for every enterprise, they do however encourage tuning the model in a way that it better suits your organization.

Some enterprises that are rapidly growing or medium-sized companies have taken inspiration from it at some level with ING Bank being one of its famous adapters.

It can be scaled and extended to multiple teams.

5.2 Structure and Roles

Role	Description
Squads	• Similar to scrum teams, a squad is autonomous, self-organizing and self-managing.
Tribes	• Multiple squads that work on the related feature area makes a tribe. A tribe may consist of 40-150 people but ideally, a tribe should have 100 individuals. A tribe has a tribe lead who is responsible for creating a productive and innovative environment for the squads.
Chapter	• At the horizontal level of the functional organization, there are chapters which are also known as the specialists. A chapter consists of individuals from different squads to be grouped into one and formed within a tribe.
Guild	• An informal group constituted of people from different tribes, who have a common interest, form a guild. A person from any squad, chapter or tribe can be a part of a guild.
Trio	• A trio is formed when for every tribe there is a design, product area, and a tribe lead.
Alliance	• Alliance – A combination of three trios makes an alliance. It is led by a product, design and a tribe lead.
Chief Architect	• A crucial member of the organization that defines the architectural vision and who also guides the designs and deals with the system architecture's dependency issues.

5.3 Planning

The squads use KANBAN, scrum sprints, XP or a mix of these agile methodologies to carry out their duties. Each squad has direct contact with the stakeholders. Face to face communication is encouraged over documentation.

Release Train dates are pre-scheduled and each team can add their part of the code whenever suitable. Operation teams act as a squad and their job is not to just deploy the code into production, but to build a structure that enables squad teams to deploy their code themselves. Development is iterative and incremental.

Tribes have gatherings on a regular basis. They have an informal get together where they show the rest of the tribe what they are working on and what they have delivered. This includes demos of working software and which tools have been used. Teams have Scrum of Scrums to discuss any roadblocks.

Feature Toggling is encouraged as you can deploy code into production, which remains hidden, that can be enabled or disabled depending on the user or the environment.

For planning, the teams use a matrix called DIBBs, which is an abbreviation for Data, Insights, Believe, and Bets. High-level planning is based on the company bets and beliefs that are supported by data and insights. Strategic alignment comes from squad level bets, that leads to tribe bets and functional bets.

5.4 Retrospectives

The agile coach conducts retrospectives while sprint planning meetings are kept optional. There is consistent communication with stakeholders and customers.

This will call for substantial changes to the organization structure.

5.5 Features

- Adapting a unique Agile Scaling Method has made Spotify achieve their goals quicker in an environment which is accommodating for every individual in the company.
- Enhanced delivery velocity.
- Emphasizes continuous integration and integrating the code in a delivery patch.
- Processes are reduced to a minimum.
- Feature toggling is helpful as it gives you an understanding of how the users will react when new features are added. You do this by rolling out a new feature to select users. Upon their feedback, you can have the opportunity to thoroughly test or make any improvements to the feature if needed, while having the simple option of toggling back to the older functionality, if need be.
- Addresses short-term challenges effectively.
- Minimized dependencies within teams.
- Lack of a firm structure makes problem-solving easier.
- Not a traditional organizational structure where the manager tells the teams what to do.
- Teams are autonomous and self-managing with minimum control.
- Promotes trust, clarity, and transparency.
- Servant leadership is practiced.
- Focuses on evaluating the team members' motivation level to ensure maximum productivity.
- Believes in learning by delivering and continuously adapting.
- Ample room for continuous improvement.
- Responds to change quickly.

5.6 Challenges in Spotify

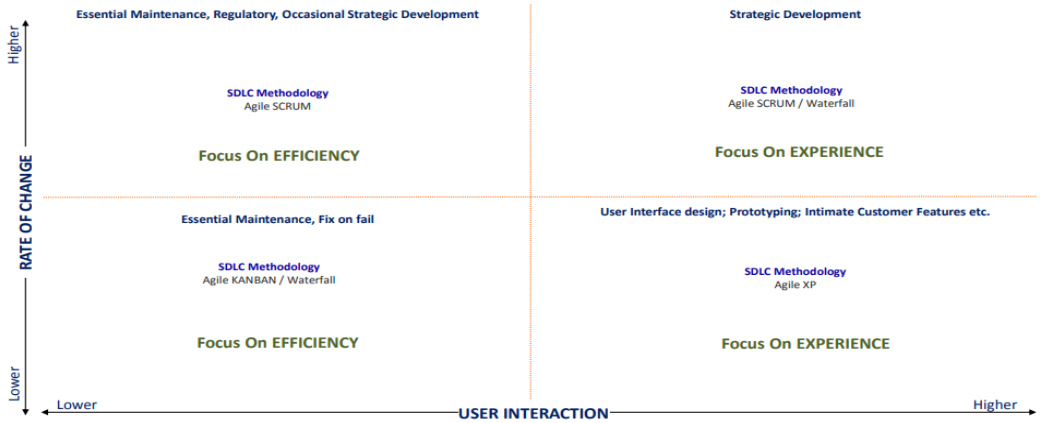
Challenge	Description
Squad alignment	A key principle for every squad in the Spotify Agile Model. Each Squad has its own mission but it needs to be aligned with the product

Challenge	Description
	strategy, and organizational priorities. To eliminate the risk of getting too much focus on the squad's mission and neglecting overall product strategy, right alignment is needed. Teams need a way to synchronize the progress of all the squads of a tribe with the product management and stakeholders. In some organisations, Chapter and Tribe Leads can make their own flex board to clarify the squads' work alignment with the common vision and strategy.
Release planning and tracking	The Spotify Agile Model is efficient in keeping the speed of the development steady thereby teams release quicker. But what if a feature is missed and never becomes a part of a release? The environment and tools should enable fast planning and clean tracking of these releases. It presents a visualization of all the planned and the actual contents of the release train.
Managing dependencies	Dependencies exist, regardless of how the Spotify agile model works to minimize it and even if the features are kept independent of each other as much as possible. Dependencies need to be visible. Squad leaders take care of problematic dependencies, especially blocking and cross-tribe dependencies. If they are not identified earlier this can often lead to reprioritization and reorganization of architectural changes or technical solutions. The supporting system should clearly show the mapped dependencies. The status for each dependency can be set and can be represented by different colours. This enhances management and visibility into the entire project. If there is a need of coordination to discuss dependencies, use a dependency mapping board. This guides the squads where the problem is. The Situation may get awry when it comes to dependency alignment among squads, chapters and tribes. To handle technical and product dependencies among tribes, we need to manage and make them work well by providing the clear picture of the threads.
Chapter lead engagement	Chapter leads meet periodically in person with their team members. This is to understand and support the team's members to achieve their goals. Leads have difficulties in obtaining a big picture of the Individual Squad product strategy and its progress. The chapter lead could only plan better and guide their team members if they are aware of their team's capacity. Especially, when chapter members are working in different squads.
Syncing for scrum of scrums	To handle dependencies and mitigate risks, scrum masters gather to resolve issues among squads. This syncing is made easy with the right

Challenge	Description
	tool where they can discuss issues by looking at the big picture supported by real-time data that is fetched from each squad's daily task management tool.
Architecture changes alignment	Tracking architectural issues is necessary. A chief architect can make a board to track architectural requirements and enforce these in accordance to the program vision. These requirements are added to the portfolio board after discussing them with the product management. In another team, system owners get the benefit of these architectural requirements and visualizing its dependencies with the product features and value stream.

6 Guidelines on the model

- Looking at the user interaction and rate of change as two parameters, we can make use of specific models. The following diagram gives an idea.



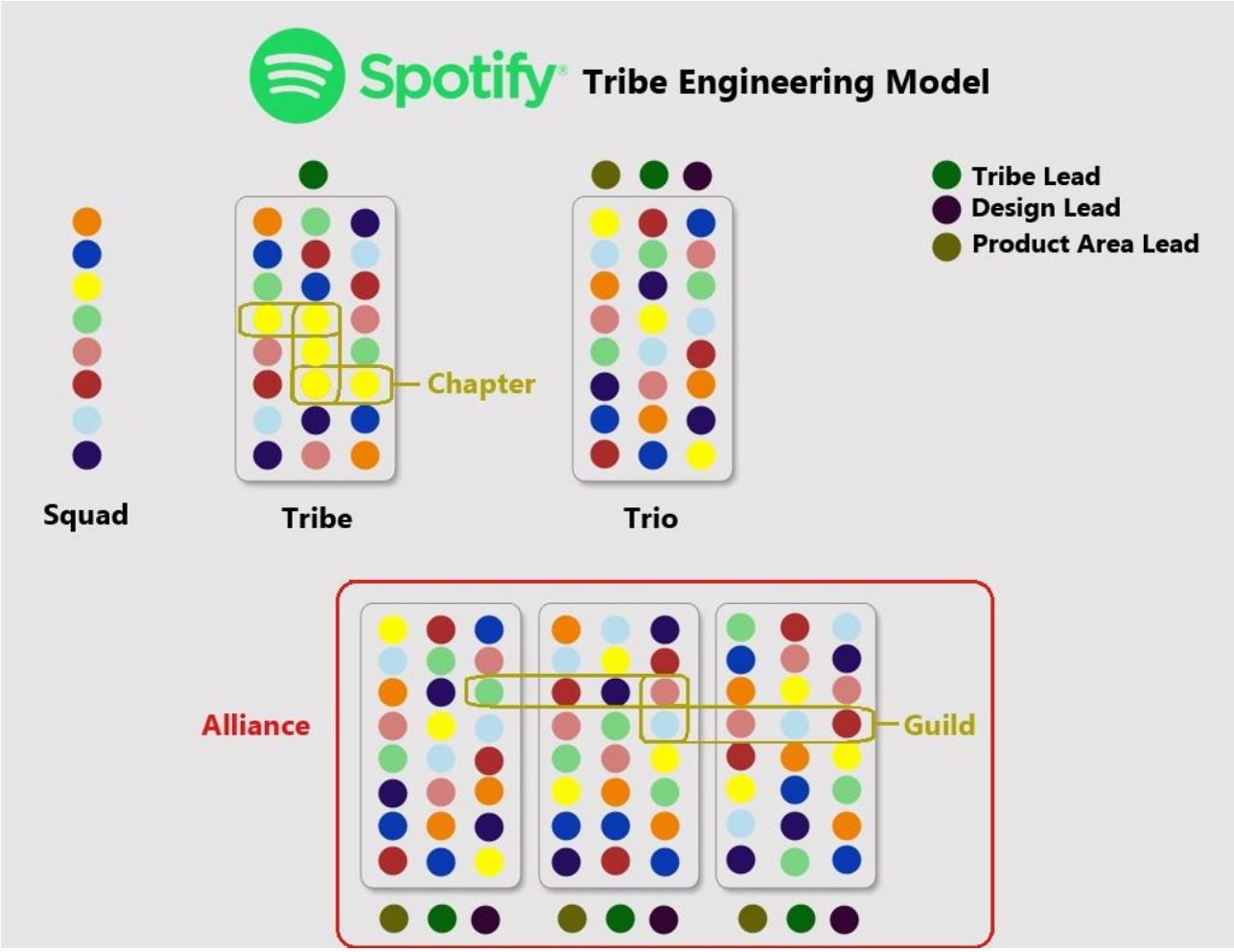
- In the table below, the considerations of application type, type of work, type of team, volatility etc. determine the applicability of different methodology.

Parameter	Description
Application Type	Strategic Maintain Run to Retire PoC
Volatility Type	High, Medium, Low
Work Type	Large/Medium enhancement Maintenance/Small enhancement Migration Large platform based

Parameter	Description
	Innovation
Variability Type	Small, Medium, Large
Team Type	Independent Frequent collaboration High collaboration
System Type	Stand-alone/Small Medium complex/Medium dependencies Highly complex/Multi-dependencies

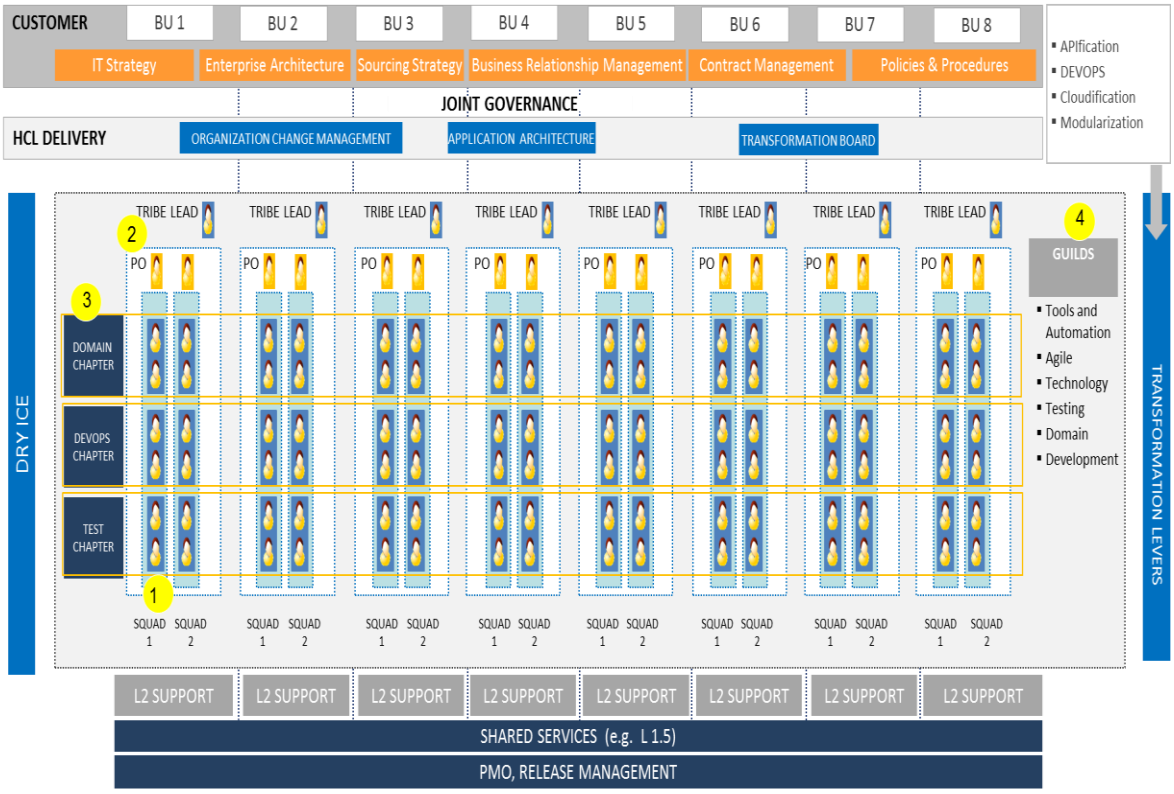
Parameter	Scrum + XP	Scrum	Kanban	Waterfall
Application Type	PoC	Strategic	Maintain Run to Retire	Maintain Run to Retire
Work Type	Innovation Platform	Large/Medium enhancements	Small changes	Maintenance Migration
Team Type	High collaboration Frequent feedback	High collaboration Frequent feedback	Independent	Independent
Volatility Type	High	Medium	Low	Medium/Low
Variability Type	Large	Medium	Small	Medium/Small
System Type	Large/Complex	Medium complex	Small / Stand-alone	Medium complex/Medium dependencies

7 Target Operating Models with Spotify



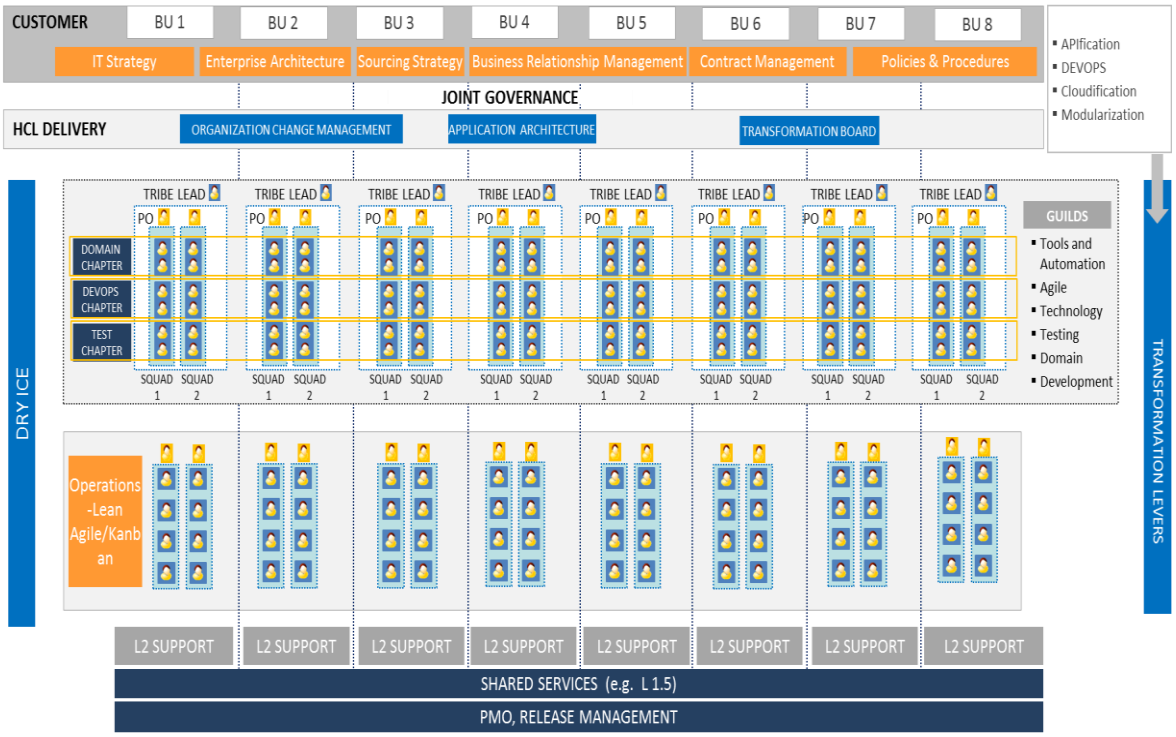
This can be depicted in our standard ToM (Target Operating Model) as follows:

SPOTIFY AGILE TARGET OPERATING MODEL

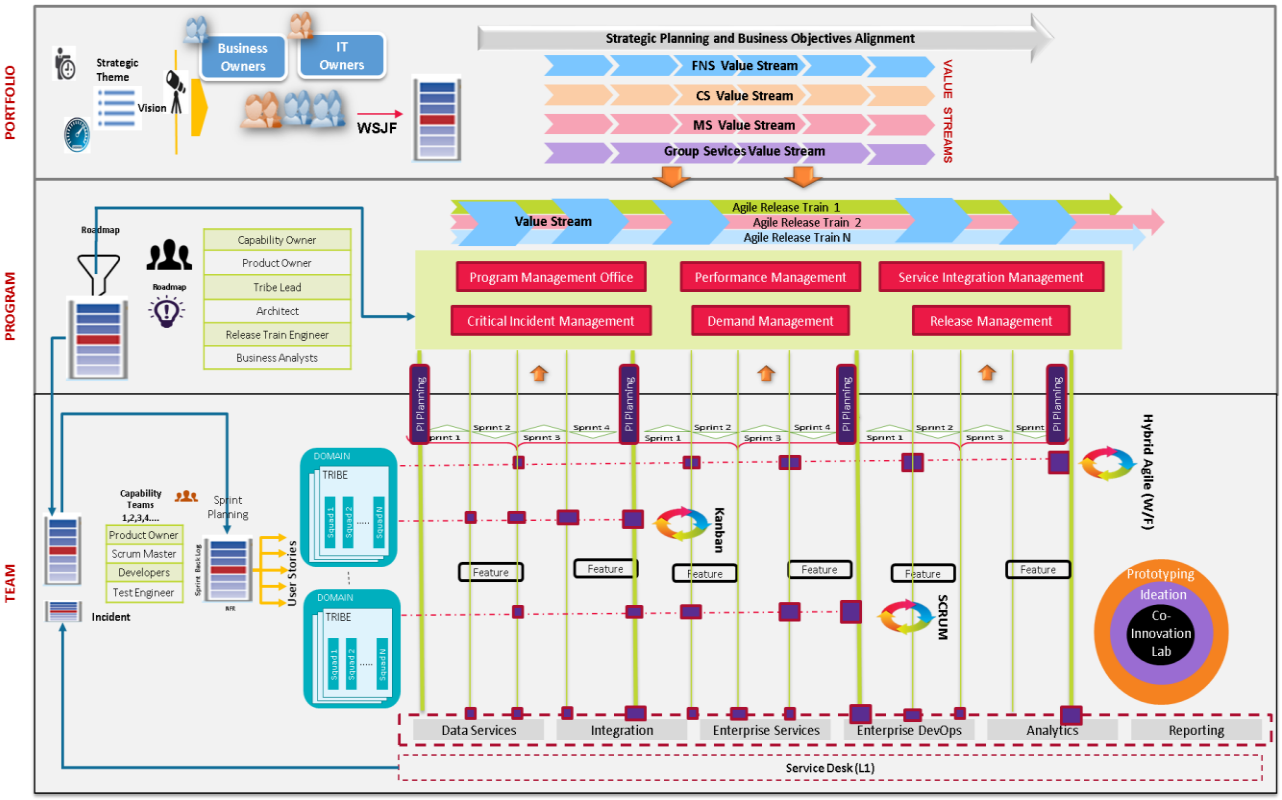


Here is a depiction of alternate model where operations team follows Kanban while rest of the organization follows Spotify Agile.

SPOTIFY AGILE TARGET OPERATING MODEL-ALTERNATE



Here is another customized version for a specific customer.



8 Factors to be considered in the solution and commercials

The solutions team need to consider the following factors and ensure their impact on delivery as well as commercials is understood.

Area	Considerations	Impacting

Transition	• Current level of agile maturity across business units (Each unit can operate in a different level)
	• Current offshoring maturity
	• Usage of distributed agile
	• Consider various scenarios like:
	  ○ Waterfall to Agile
	  ○ Agile to Agile.
		• Absorption: Include pair programming and architectural runway (Existing code, hardware components and software functionality that technically enable near term business features)
		• Replication: Sprint based execution
		• Boot camp duration and costs (focus on Agile foundation, Agile Associate, Scaled Agile. behaviour aspect - People mind set, culture
		• open-up and express their view)
		• Include time and effort for doing FENIX based assessment of the in-scope portfolio
		• Onshore ratio
		• Timeline

Steady State	• Agile maturity of different business units • Level of life cycle management and DevOps tools usage • Type of full stack engineers required • Availability of roles like PO, Agile coach, Scrum master with the customer • Time proposed for transformation • Quantum of work • Current effective tools usage • Usage of DevOps • Current mapping of resource levels	• Operating Model • Evolution of ToM across the 3 stages Standardize, Optimize and Transform and time line • Squad size • Release cycle • Onshore ratio across months/years as the customer moves up on the maturity curve • Type of resources and their distribution • Productivity % resulting from people, process and tools
Commercials	• Physical Infrastructure • Connectivity requirement considering more online interactions / meetings	• # Travels (could be more – ING experienced this very significantly)

Area	Considerations	Impacting
	• Typical squad size per ODC (there could be many special requirements per squad)	• Special infrastructure (meeting rooms, furniture, boards, interactive rooms etc.) costs • Special equipment like monitor (big size, two monitors per person) • Additional tools/license ILF (initial License Fee) and RLF (Recurring License Fee) • Cost of baselining the resources of the customer and plotting them against SFIA or Dreyfus model • Recruiting costs (cost of running hackathon) • Cost of hosting customer periodically • Cost of building Academy • Cost of configuration/customization of tools like Kalibre, HawkEye for the engagement • Training / Upskilling costs covering the initial training required to make the resources more engagement ready as well as upskilling as they move the Dreyfus cycle. The traditional training materials will not do.