
accenture technology
Agile University: Delivery School
Scaling and Distributed Delivery
High performance. Delivered.
Strategy | Digital | Technology | Operations

Agenda

Accenture Distributed Delivery Models & Considerations

Multiple Vendors - Communication Patterns

Scaling Considerations & Practices

Best Practices

Introduction

Agile Recommends

- Small teams: (7-10 team members)
- Co-located Team
- Onsite Customers

Does this mean that Agile cannot be scaled or distributed?

Distributed Agile ?

Distributed Agile is where :

- Groups/teams working on a shared-goal but located in different geographies
- The sponsors, end-users and the development teams operating from different locations

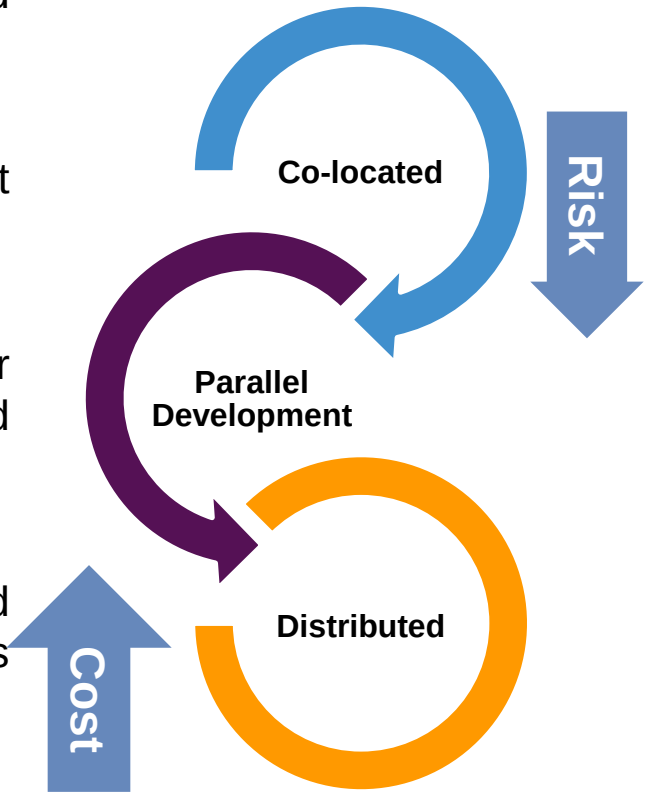
Challenges in Distributed Agile

- **Planning / Tracking and Work Distribution**
- **Team Organization** across locations
- **Cultural Differences** within the teams across locations
- **Communication & Collaboration** across the teams
- Teams in **different Time Zones** across the globe
- **Lack of Face to Face** availability of Product Owner and Business Users with the Project Team
- **Multiple Sign-off** layers, delaying development

Accenture enables the right people at the right locations at the right time

Accenture Operating Models provide a set of options on how to organize and rotate Agile teams for **co-located** and **distributed** types of Agile projects:

- No solution fits all: the **right roles** in the **right locations** at the **right stage** of software development is dynamic.
- **Rotation** of key roles between locations is critical for establishing relationships and trust between distributed teams.
- Understanding the impact that **team distribution** and **dynamics** have on the overall project is key at early stages and should not be left until the project starts.



Distributed delivery Model 1 – Co-located

Interaction Style

- Co-located team
- Team works directly with the Product Owner/Business users

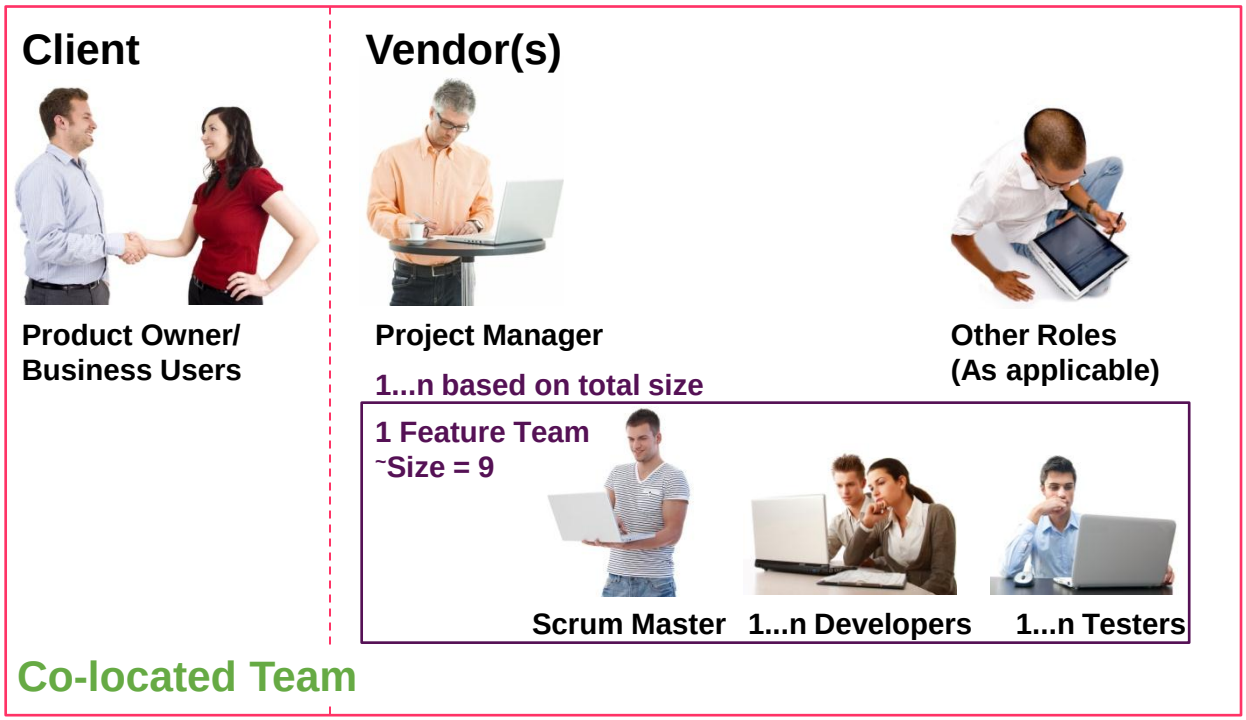
Selection Criteria

- Business requirements are too critical or confidential to be outsourced outside customer's location
- Team has limited knowledge of business functionality
- Limited availability of Business Users/Product Owner
- Adequate budget

Team Structure

- **Onshore:** Business Users, Product Owner and Delivery team (Project Manager, Designers, Developers, Configurators, Testers, Architects, Scrum master).

All roles are located at one place.



Distributed delivery Model 2 – Offshore Delivery Model

Interaction Style

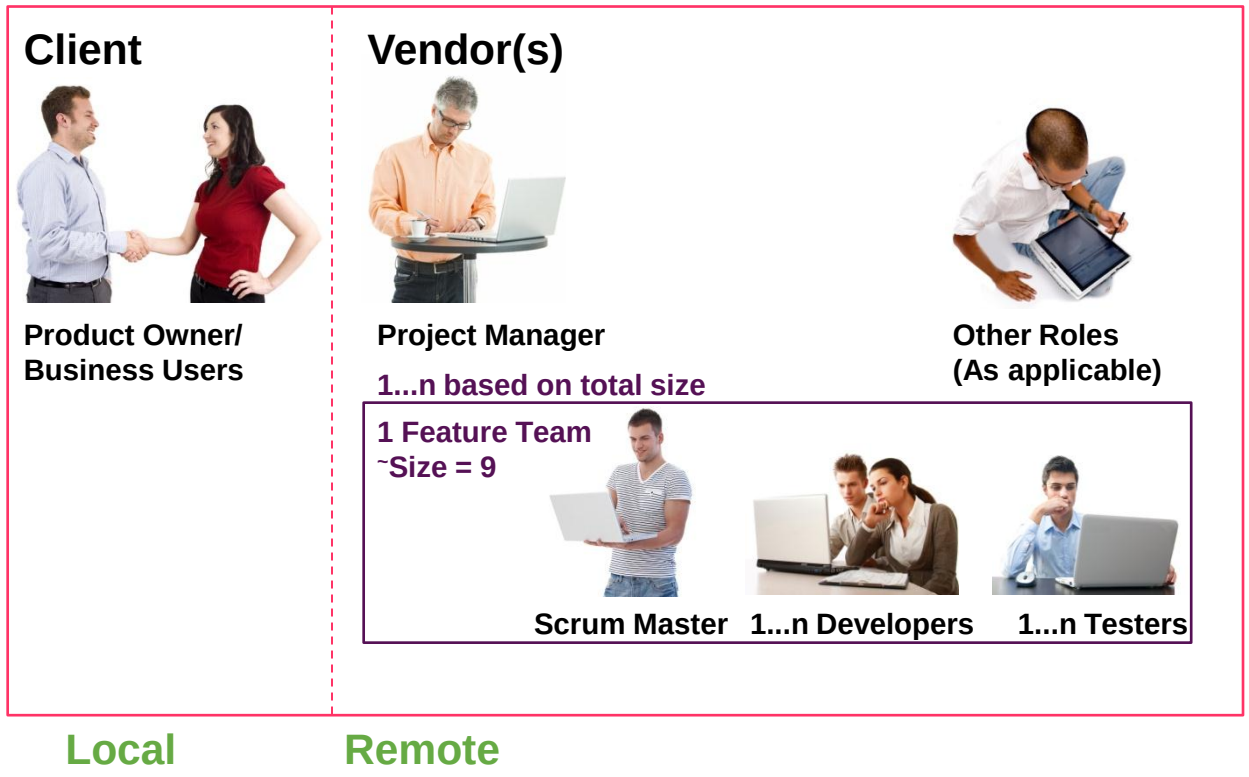
- Distributed team
- Team works directly with the Product Owner/Business users

Selection Criteria

- Simple Business requirements
- Team has good knowledge of business functionality
- High availability of Business Users/Product Owner

Team Structure

- **Onshore:** Business Users and Product Owner.
- **Offshore:** Delivery team (Project Manager, Designers, Developers, Configurators, Testers, Architects, Scrum master).



Distributed delivery Model 3 – Distributed delivery model with offshore development

Interaction Style

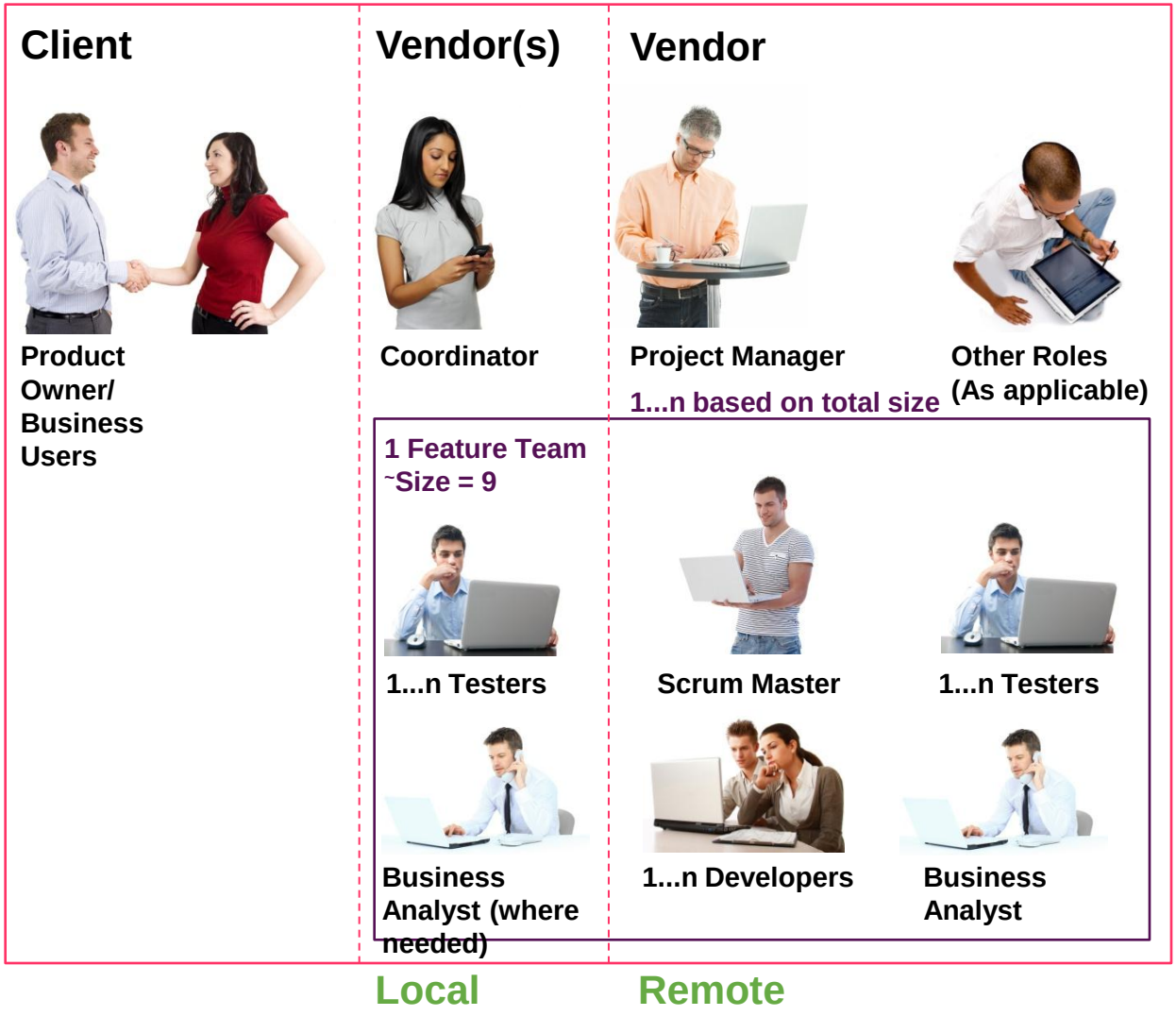
- Business Analyst/Quality Analyst(s) work directly with Product Owner
- The remote site team communicates through Business Analysts for the requirements and directly for the review

Selection Criteria

- Business requirements are complex or not very well-defined
- Remote site team has limited knowledge of business functionality
- Limited availability of Business Users/Product Owner

Team Structure

- **Onshore:** Business users, Business Analyst, Product Owner, Testers, Functional Designers (for Packaged development)
- **Offshore:** Development & Testing team (Project Manager, Designers, Developers, Configurators, Testers, Architects, Scrum master) , Business Analyst



Distributed delivery Model 4 – Distributed delivery model with parallel development

Interaction Style

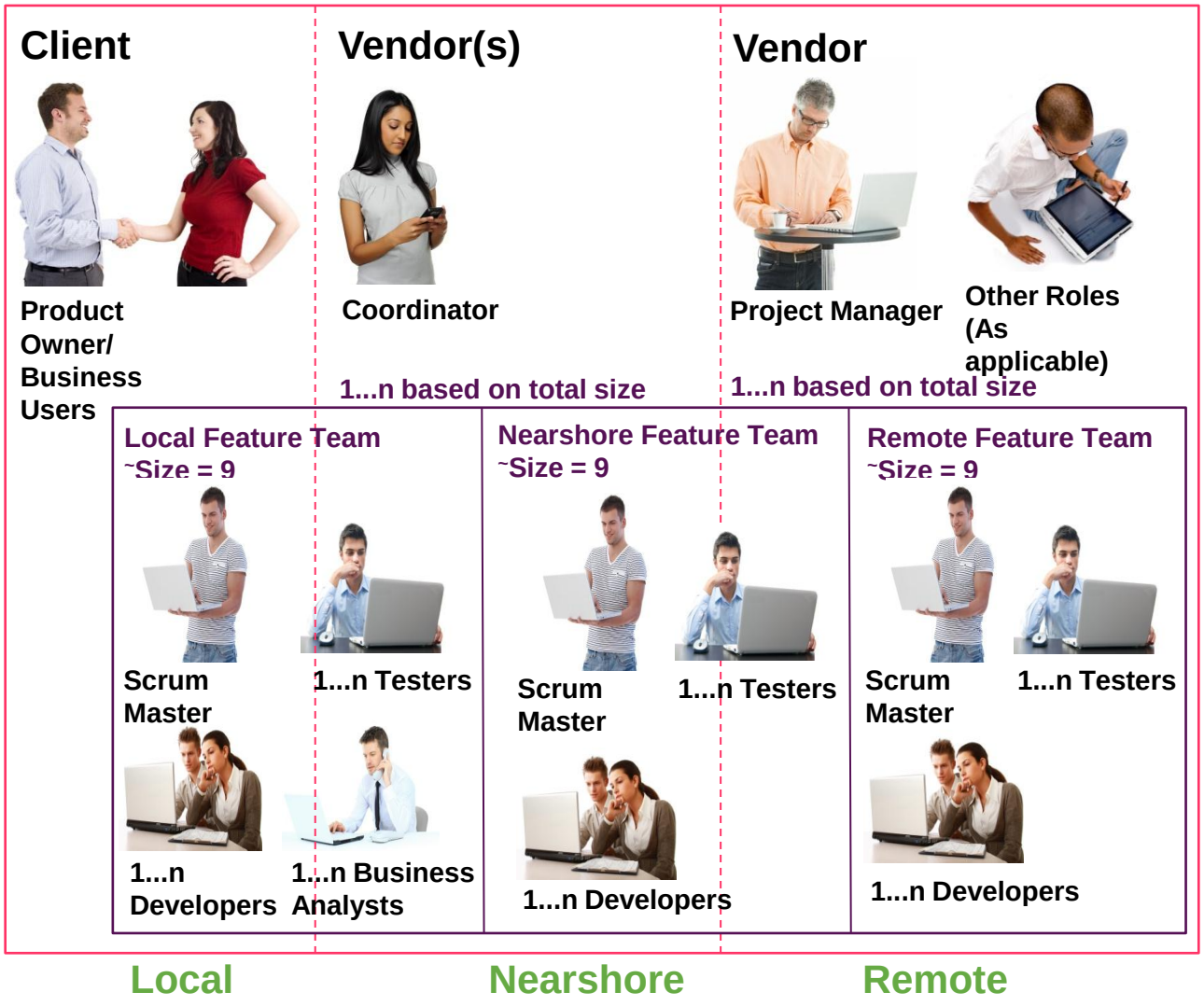
- Separate Scrums conducted locally as well as remote site
- The remote site team communicates with Business Analyst to understand the business requirements. Scrum Master can be at both locations.
- Teams need high level of tooling support (collaboration platforms)

Selection Criteria

- Business requirements are complex and not well-defined
- Remote site team has limited knowledge of business functionality
- Limited availability of Business Users/Product Owner
- Development of some application areas cannot be shifted to remote locations

Team Structure

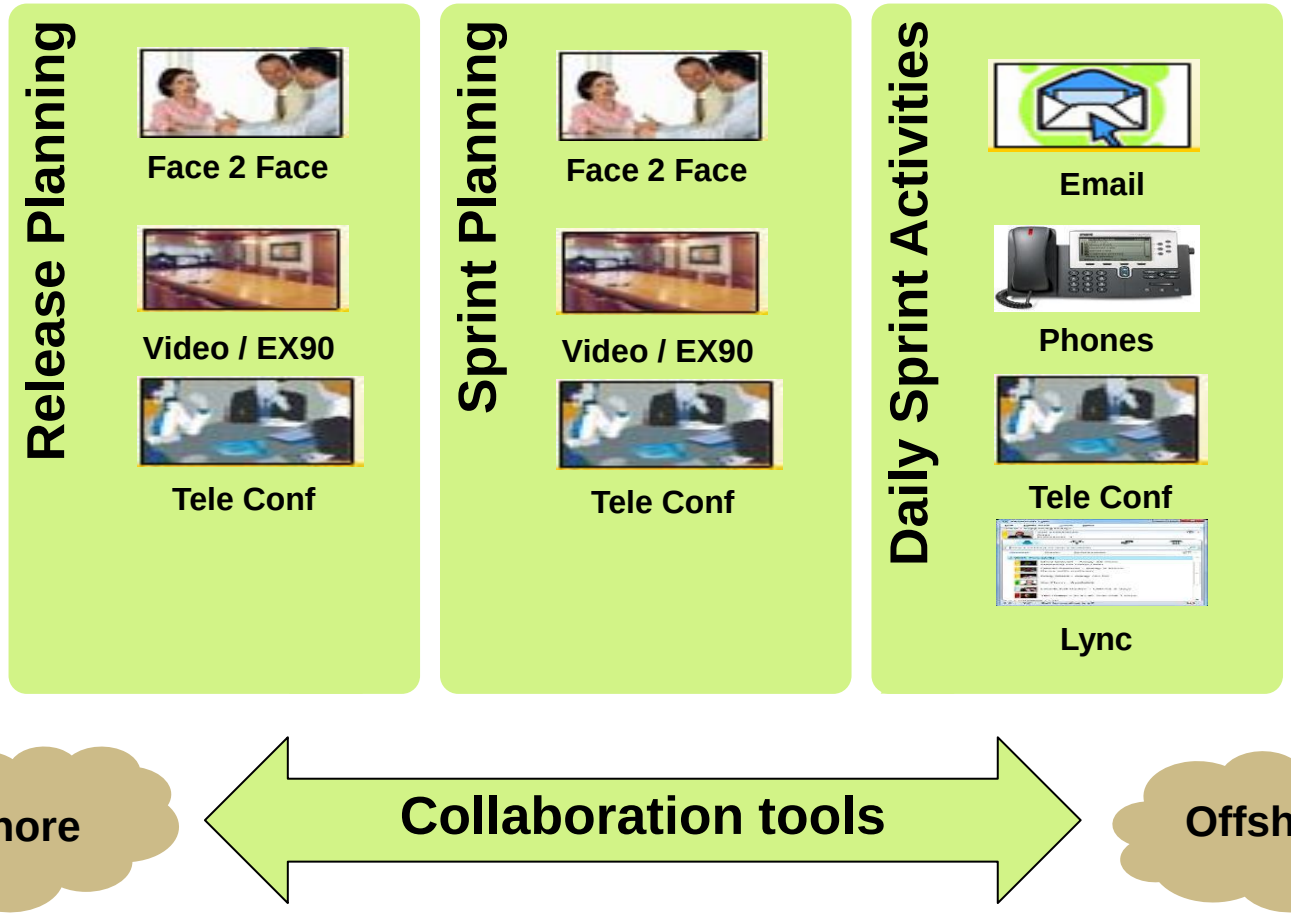
- **Onsite:** Business users, Product Owner, Business Analysts and Development team (Project Manager, Designers, Developers, Configurators, Testers, Architects, Scrum master)
- **Offshore:** Development team (Project Manager, Designers, Developers, Configurators, Testers, Architects, Scrum master)



Communication & Collaboration Practices

- Ensure there is the **right degree of documentation**
- Conduct daily inclusive **onsite/offshore stand-up** meetings
- Conduct frequent client meetings and **demos**
- Ensure effective usage of **collaboration tools**
- Provisioning for presence of an **onsite coordinator** and **client presence at offshore**
- Frequent **status reporting** to stakeholders
- Ensure **Retrospectives** are held for the teams to reflect on communication and collaboration issues and create actions to address them

Communication Infrastructure for Collaboration

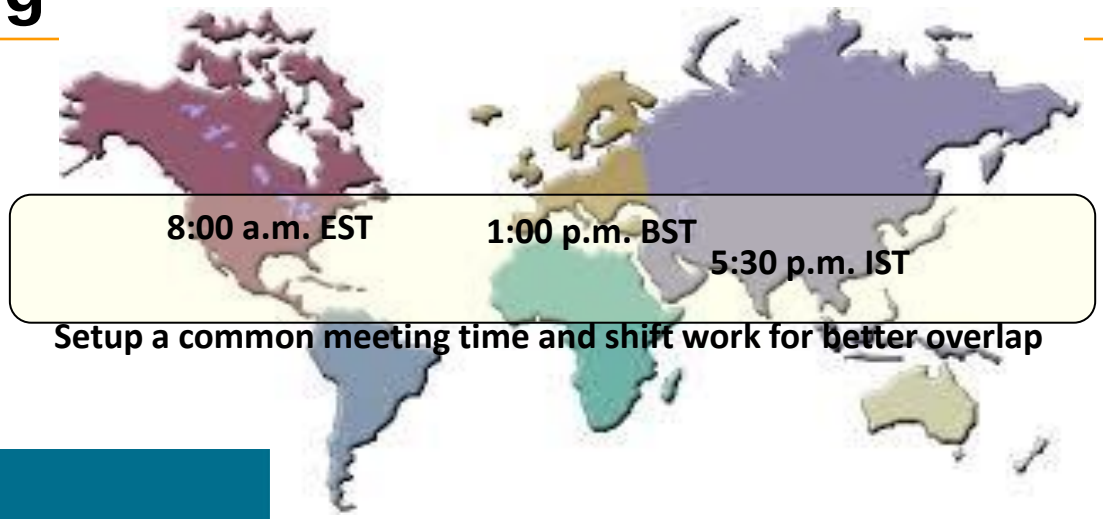


Cultural Differences

- Greater **empowerment** of teams
- **Rotation of coordinators** between locations
- **Frequent visits** by onshore team members to offshore and vice versa
- Organizing various **social events** and fun activities for team bonding
- Circulate photographs or videos of team members so people can match a face with a name

Cross-Culture Training

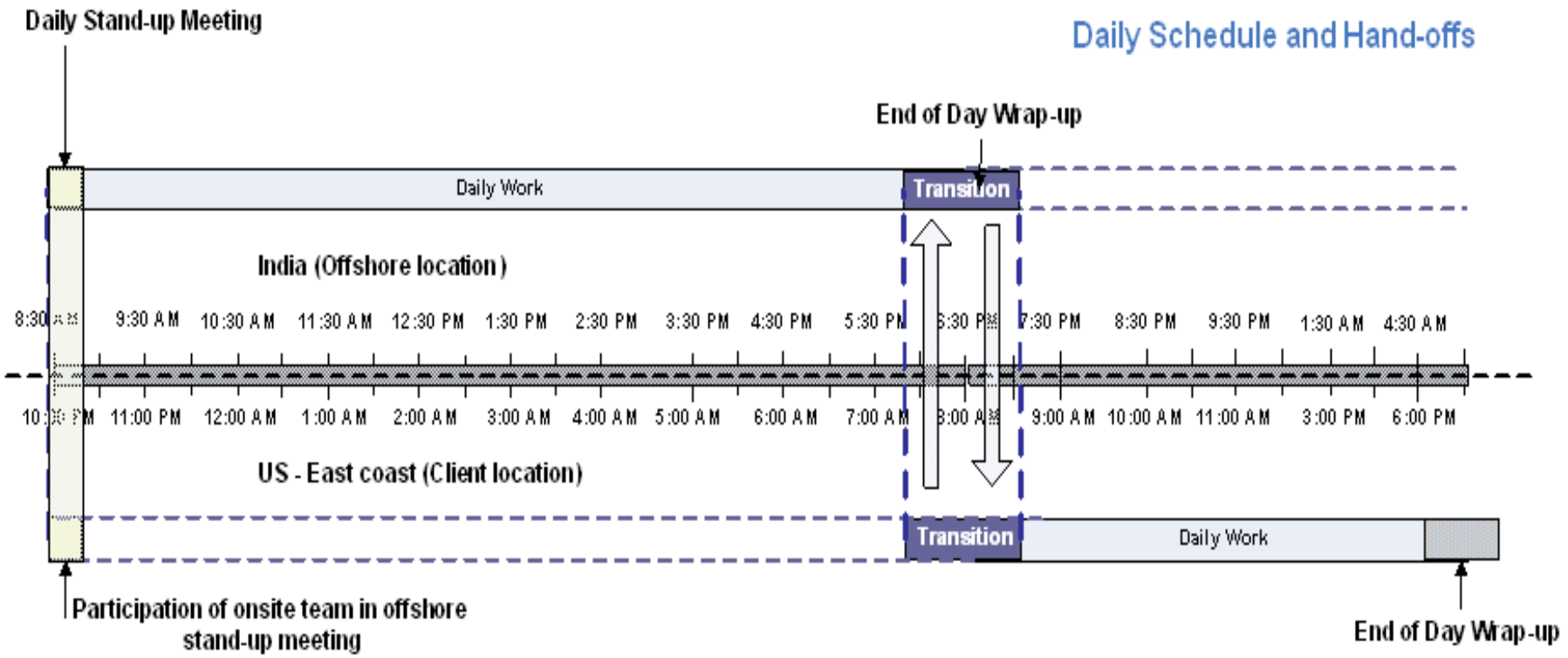
Getting Teams across the globe to work together



Setup a common meeting time and shift work for better overlap

Day in Life of an Agile Project - Overlap

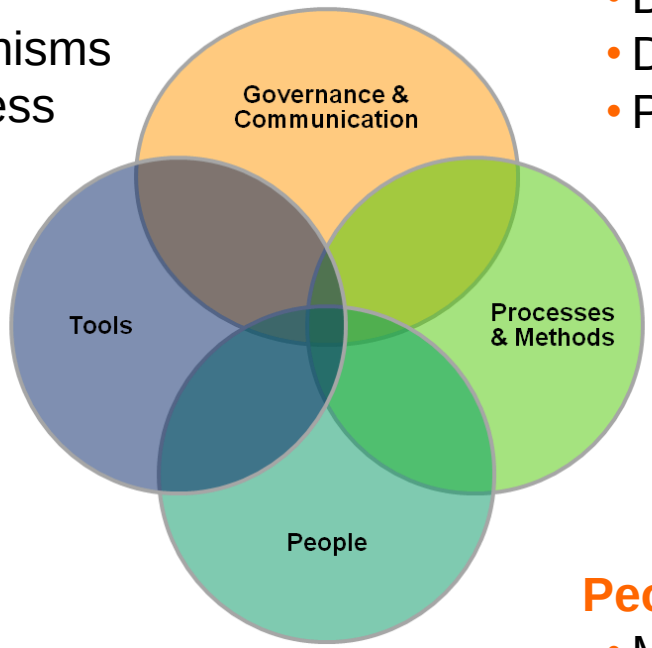
- Combined stand up at start of the day
- Transition meetings immediately after that where required
- During closing hours another transition meeting for wrap up
- Certain locations may have to adjust the office timings to ensure sufficient overlap



Other Considerations in Distributed Agile

Governance & Communication

- Sponsorship
- Governance Model
- Communication Mechanisms
- Business Responsiveness



Processes & Methods

- Requirement Nature
- Distributed Planning
- Distributed Estimating
- Project Management

Tools

- Collaboration enablers
- Configuration Management
- Continuous Integration

People

- Mindset
- Functional skills offshore
- Agile Experience
- Agile Training/Coaching

Agenda

Accenture Distributed Delivery Models & Considerations

Multiple Vendors - Communication Patterns

Scaling Considerations & Practices

Best Practices

Introduction to Multi-Vendor Environment

Today's enterprise has moved strategically from single-vendor environment to multi-vendor environment.

Key Characteristics

- Teams are formed from different vendor organizations
- Differences in Processes and Practices
- Different areas and levels of expertise
- Multiple cultures within each vendor organization

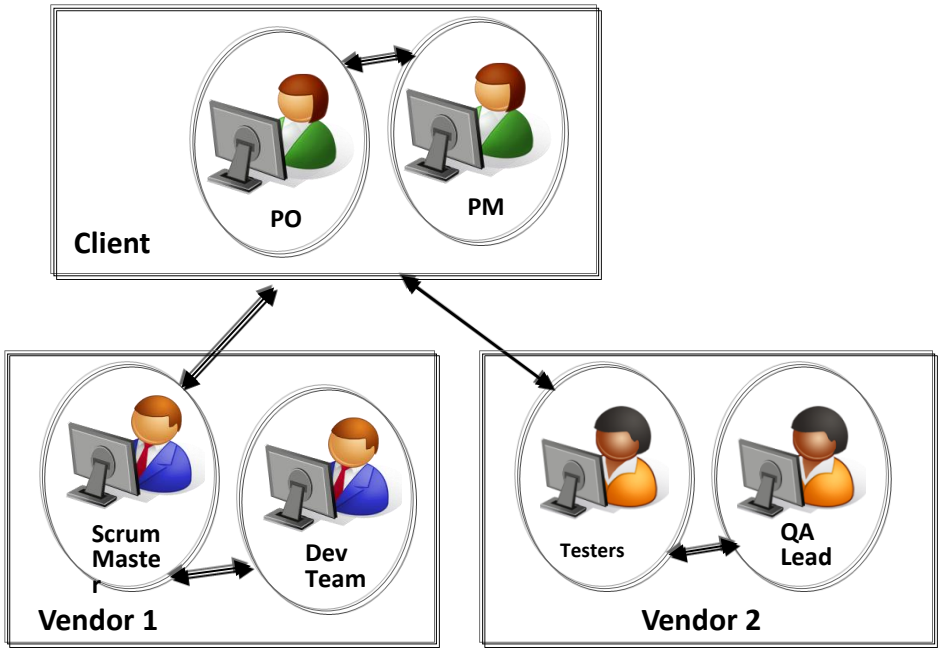
Benefits

- Experience the best of breed solutions
- Optimize performance
- Achieve reduced risks and cost
- Ensure specialized support
- Experience flexibility of choice

Multi-Vendor Models - Customer Managed

Criteria:

- Business requirements are Complex and / or Confidential
- Only one vendor owns the development work
- Client controls the coordination and communication between vendors



Benefits:

- Closer collaboration with the Client
- Vendors obtain first hand information
- Less potential gaps in communication
- Decision making process is faster
- Clients can benefit from centralized monitoring and tracking

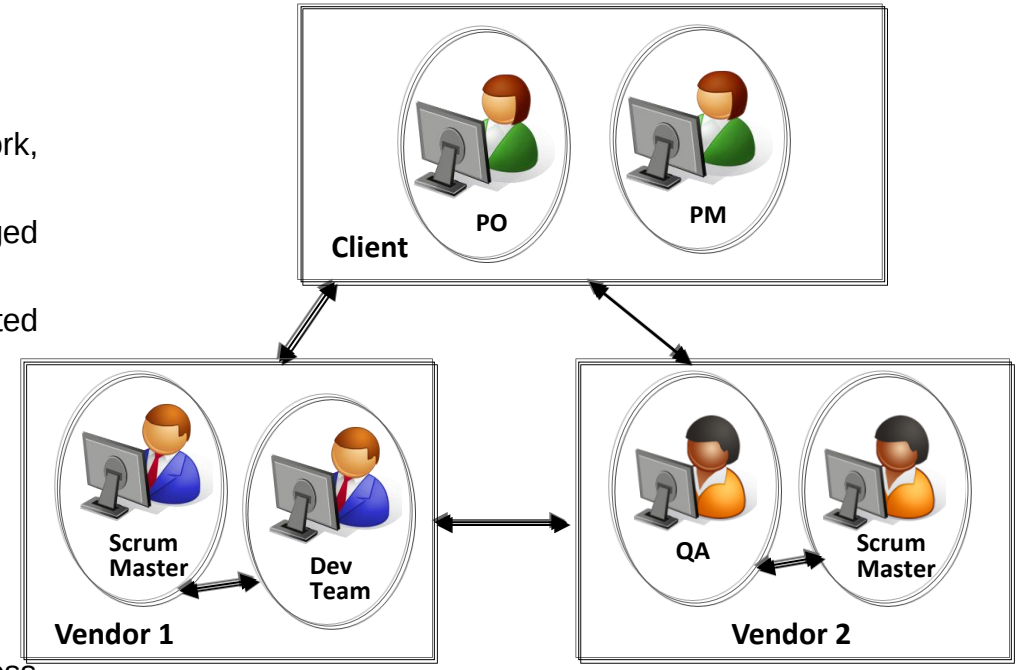
Challenges:

- More Client involvement is required throughout the project life cycle
- Potential delays in information shared between vendors
- Milestones are driven by the Client which can lead to delays in arriving at a consensus between vendor teams

Multi-Vendor Models - Vendor Collaboration

Criteria:

- Business requirements are Simple
- Only one vendor owns the development work, the other is QA
- Coordination and communication is managed by the vendors
- Only the critical transactions are coordinated by the Client



Benefits:

- Less time spent by the Client
- Faster response to change
- Risk mitigation is improved
- Decrease in communication delays across vendor teams

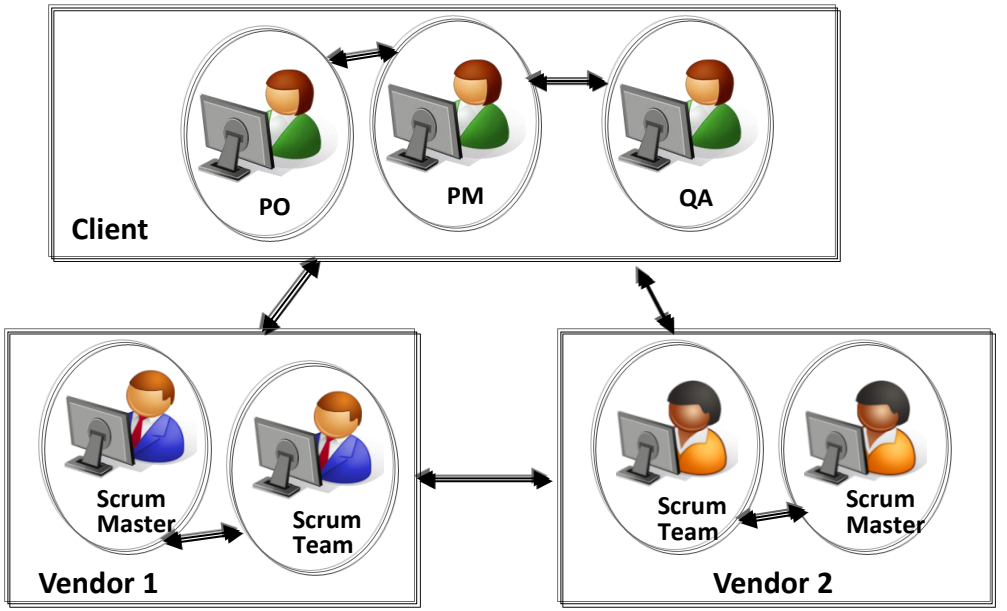
Challenges:

- Decision making may be slower
- Good leadership and communication skills are required from vendors
- Getting a consolidated dashboard may be time consuming (monitoring and tracking is done by individual vendors)

Multi-Vendor Models - Parallel Scrums

Criteria:

- Business requirements are Complex but less dynamic
- Individual vendor teams have well-defined and mature processes
- Development can be owned by multiple vendors
- Coordination and communication is managed by the vendors
- Only the critical transactions are coordinated by the Client



Benefits:

- Ownership of the individual components is clear and distinct
- Issues arising out of cultural difference is controllable since the coordination with client is limited to few vendor stakeholders
- Milestones are decided and agreed upfront by individual vendor teams bringing consensus forward

Challenges:

- Strong Continuous Integration environment is required
- Management skills are required by individual vendor teams
- Integration risk is higher as it is conducted at a later stage of the project by the Client teams

Agenda

Accenture Distributed Delivery Models & Considerations

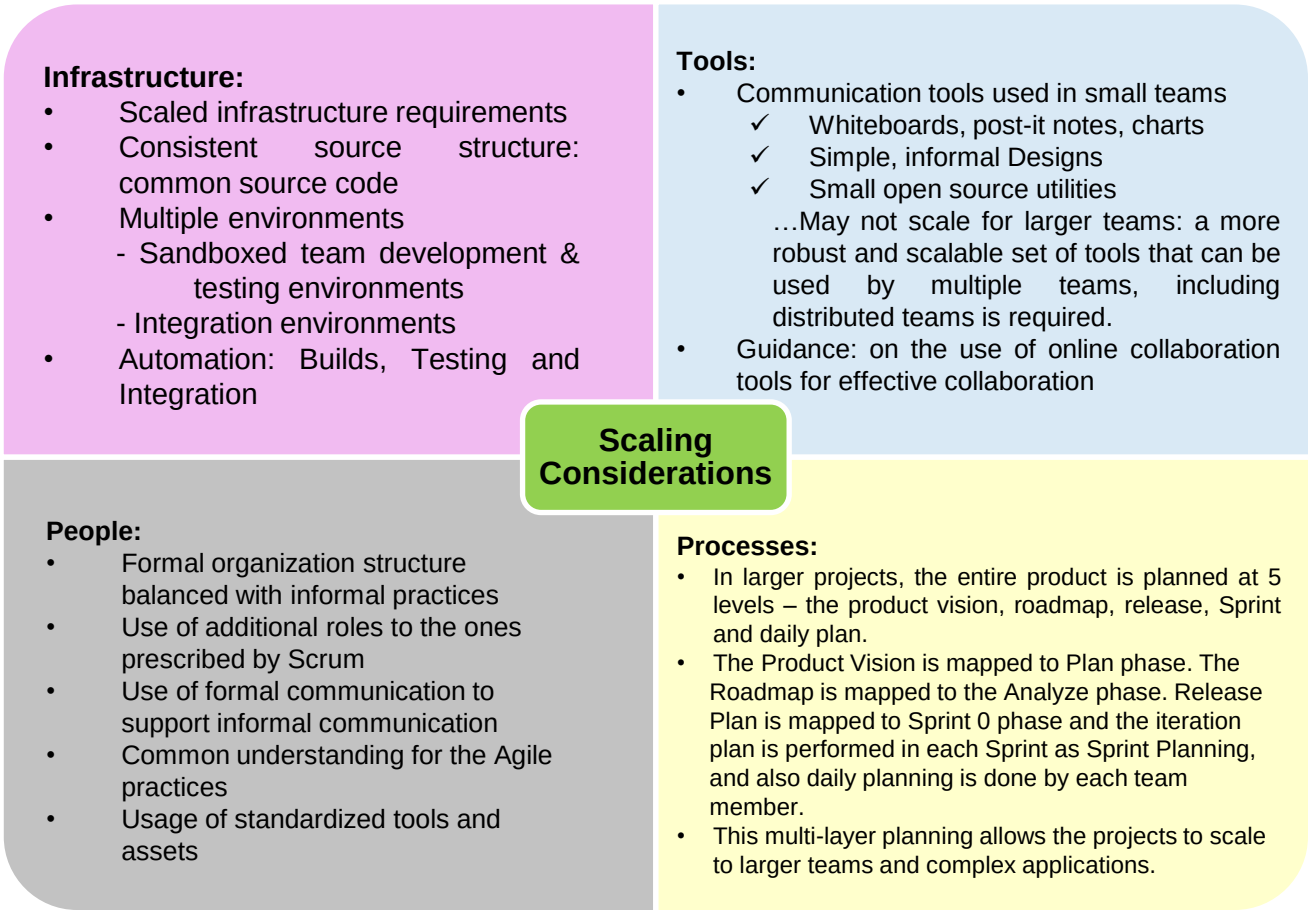
Multiple Vendors - Communication Patterns

Scaling Considerations & Practices

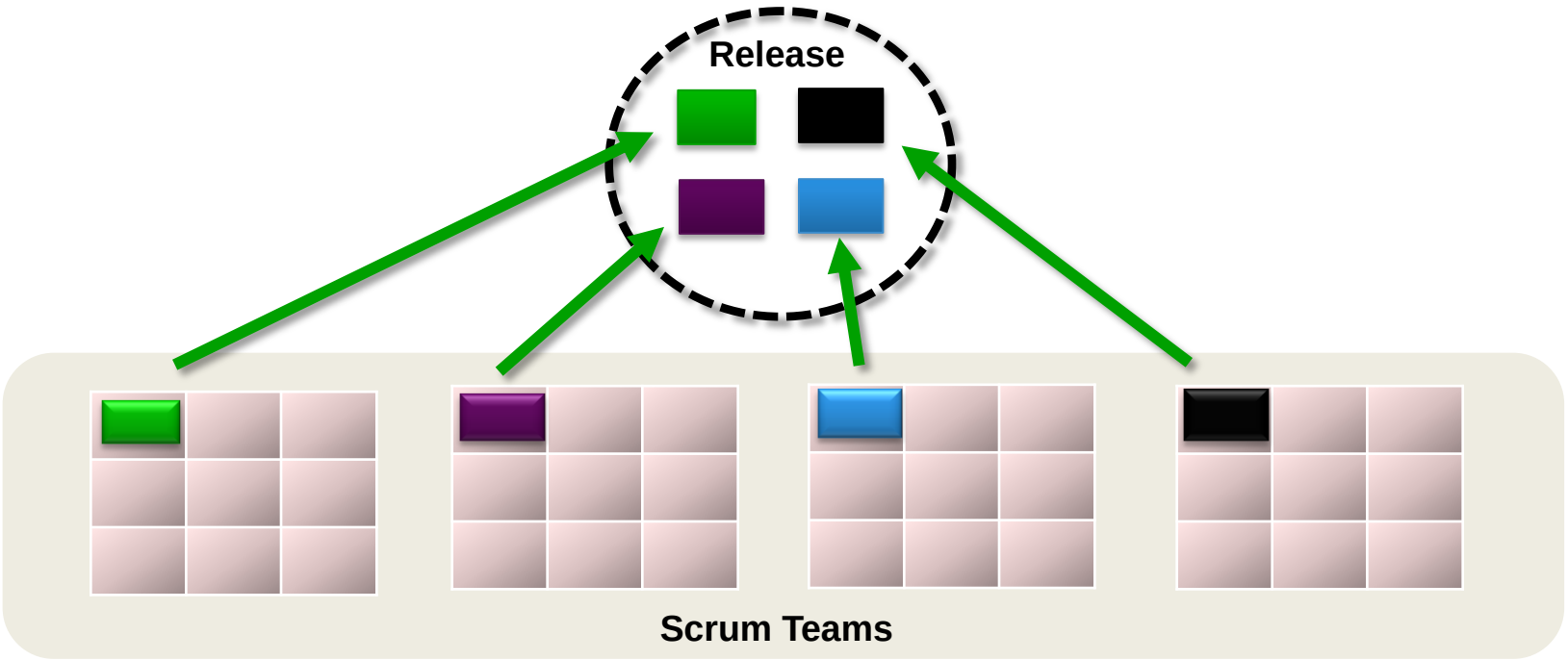
Best Practices

Scaling and its Considerations

While launching a large-scale agile project that requires more than a couple of Scrum teams then careful consideration should be given to how you scale the management of the project.

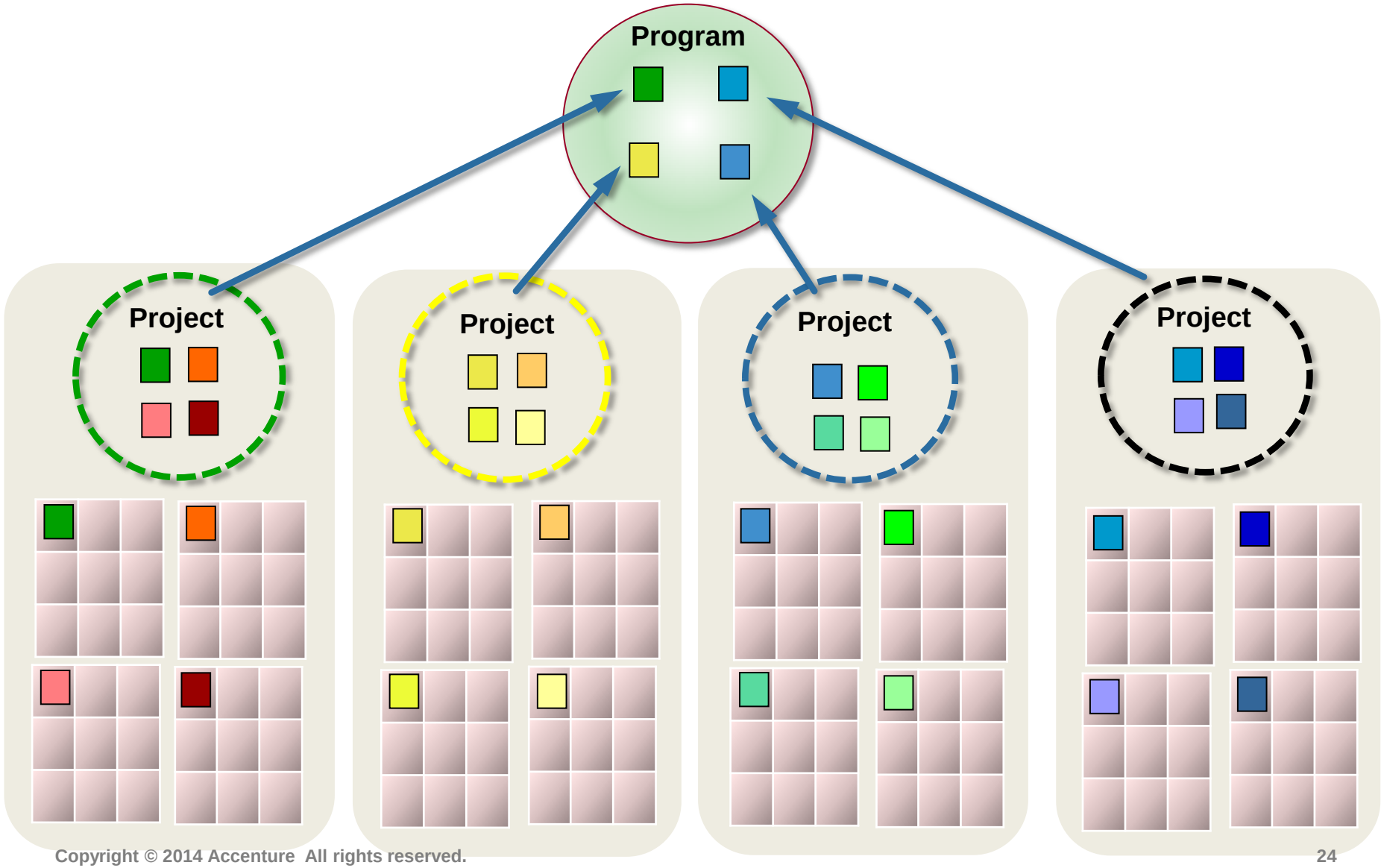


Scaling Practices – Scrum of Scrums



Scrum of Scrums Meeting are attended by Scrum Masters or senior members of each team

Scaling Practices – Scrum of Scrum of Scrums

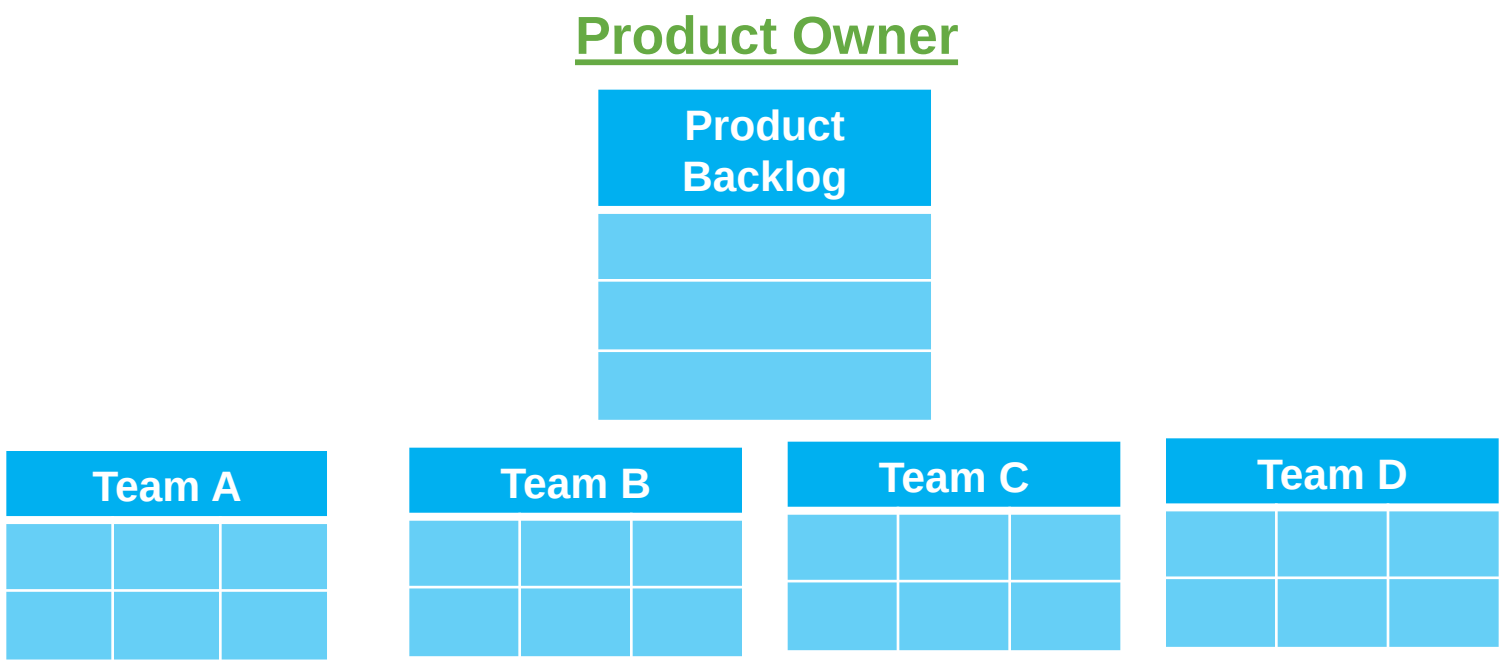


Scaling Product Backlog

If the project team size is large ~50 People then we can take three approaches to scale the Agile Project

Approach 1 - When building something where all the modules are intimately integrated, a single, tightly managed, master backlog may work well.

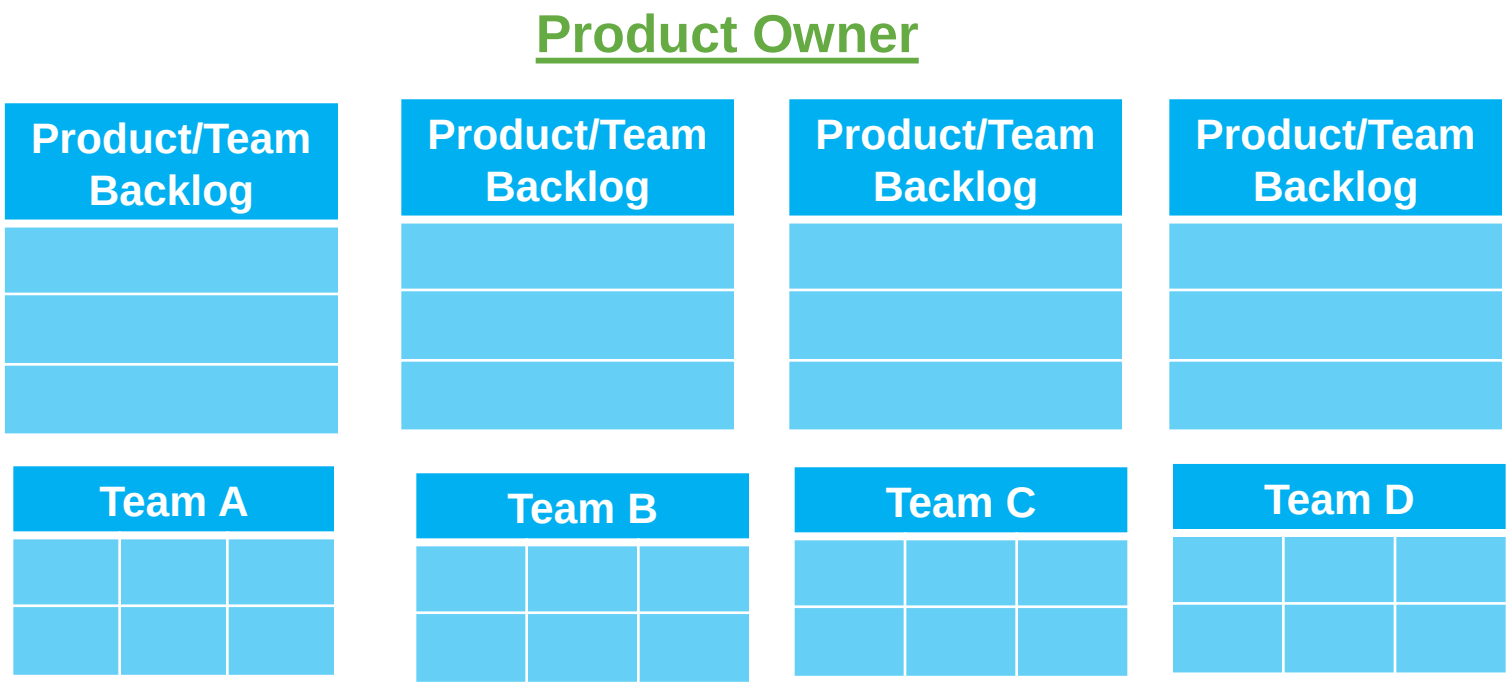
- Divide project team into multiple teams and they can work from one Product Backlog
- One Product Owner will be responsible for that Product Backlog



Scaling Product Backlog

Approach 2 – While working on different technical implementations for the same product.

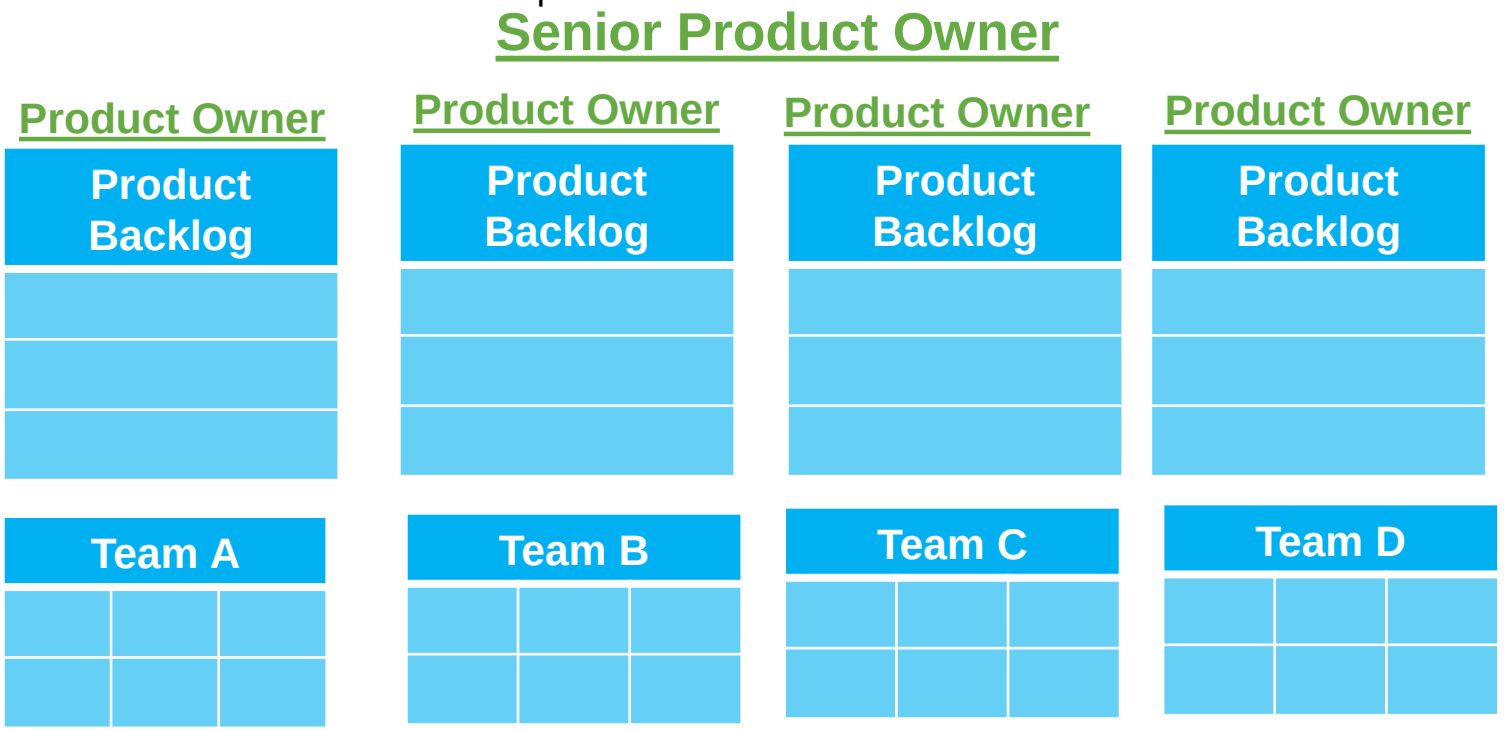
- Divide project team into multiple teams
- Assign one Product Backlog to each team
- One Product Owner will be responsible for all Product Backlogs



Scaling Product Backlog

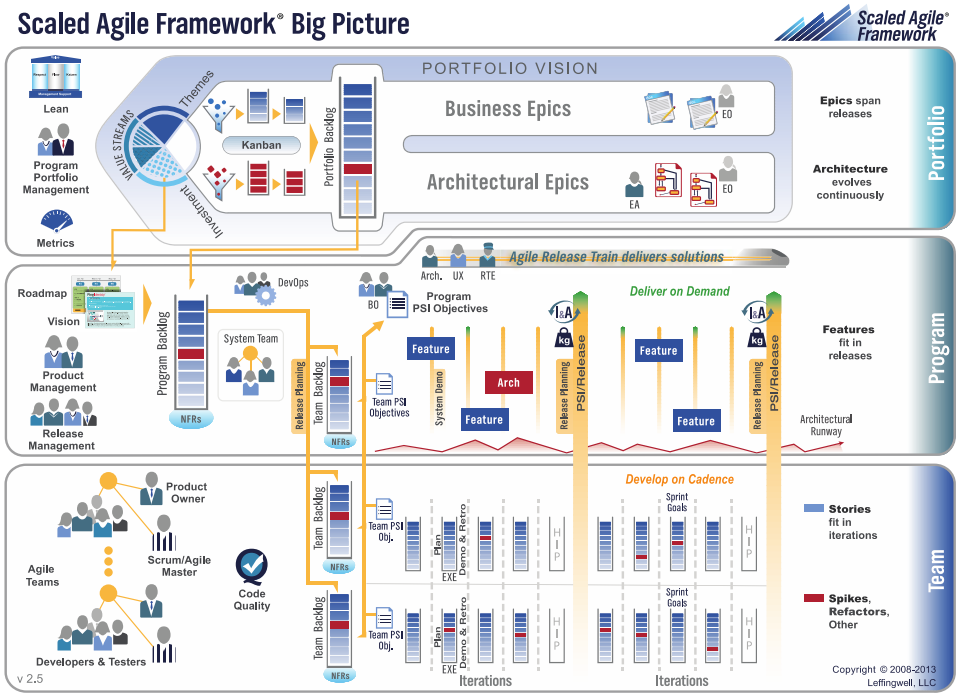
Approach 3 – When teams are working on different functional modules that are independent of each other with different functional leads

- Divide project team into multiple teams
- Assign one Product Backlog to each team
- Assign one Product Owner to each Product Backlog
- All Product Owners will report to one senior Product Owner



Scaled Agile Framework – Scaling Agile to Enterprise Level

The Scaled Agile Framework® (pronounced SAFe™) is an interactive knowledge base for implementing agile practices at enterprise scale. It is a proven, publicly-facing framework for applying Lean and Agile practices at enterprise scale.



CORE VALUES

1. Program Execution
2. Alignment
3. Code Quality
4. Transparency

Synchronizes alignment, collaboration and delivery for large numbers of teams

http://www.scaledagileframework.com/

Agenda

Accenture Distributed Delivery Models & Considerations

Multiple Vendors - Communication Patterns

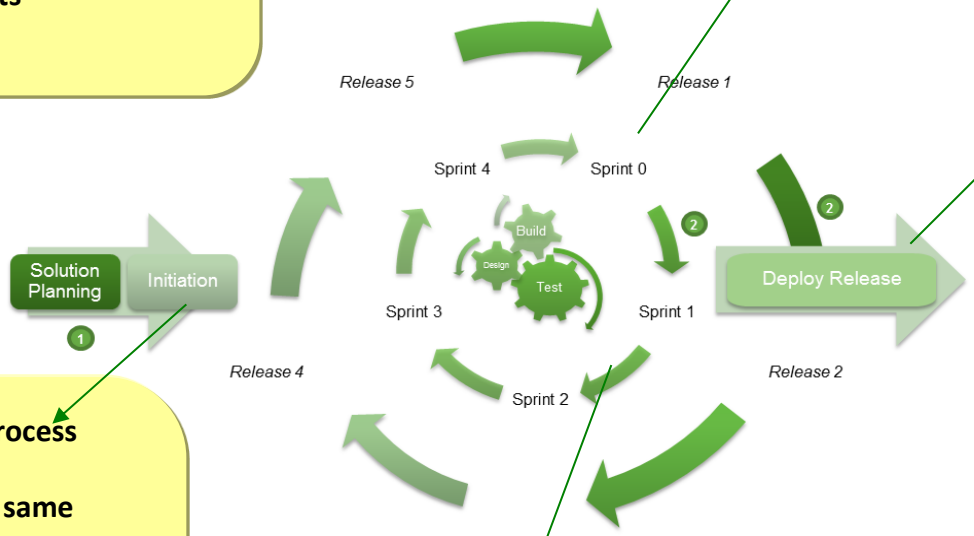
Scaling Considerations & Practices

Best Practices

Some of the Best Practices

- Ensure contracting guidelines for Agile are consulted
- Consider infrastructure and communication overheads for estimation
- Adopt the operating model which is best suited for the requirements

- Conduct a joint release planning with all the vendors
- Consider cross-vendor dependencies while deciding the priorities and estimates
- Consider Architectural and Testing dependencies while planning



- Ensure that there is a common integration environment and it is accessible by all the vendors
- Conduct cross-team review for the integration test scripts
- Common defects log which is shared across vendors

- Conduct / participate in a process alignment workshop
- Ensure all vendors have the same understanding of requirements. Conduct a workshop if required
- Setup a robust change management process
- Setup the required infrastructure and test it

- Conduct Scrum of Scrums with the different vendors involved
- Joint Sprint Planning to plan the test cycles, if done by different vendors
- Agree on a common definition of Done
- In case of design dependencies across vendors, ensure that a common design is agreed and shared
- Ensure that a common set of metrics is tracked

Questions & Answers