



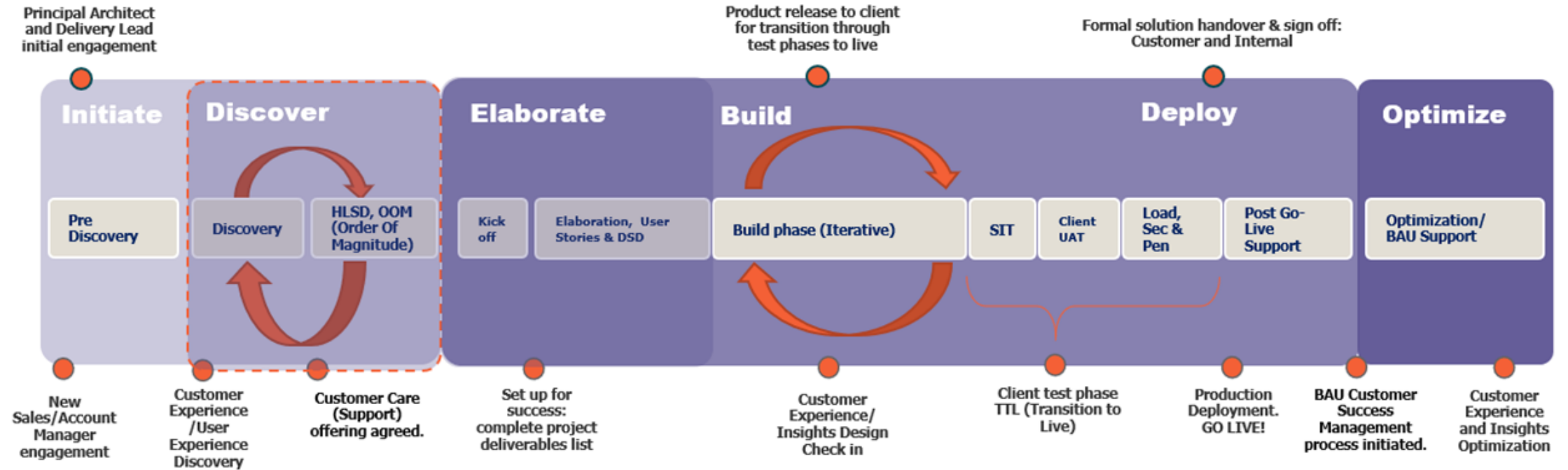
TEMENOS

THE BANKING SOFTWARE COMPANY

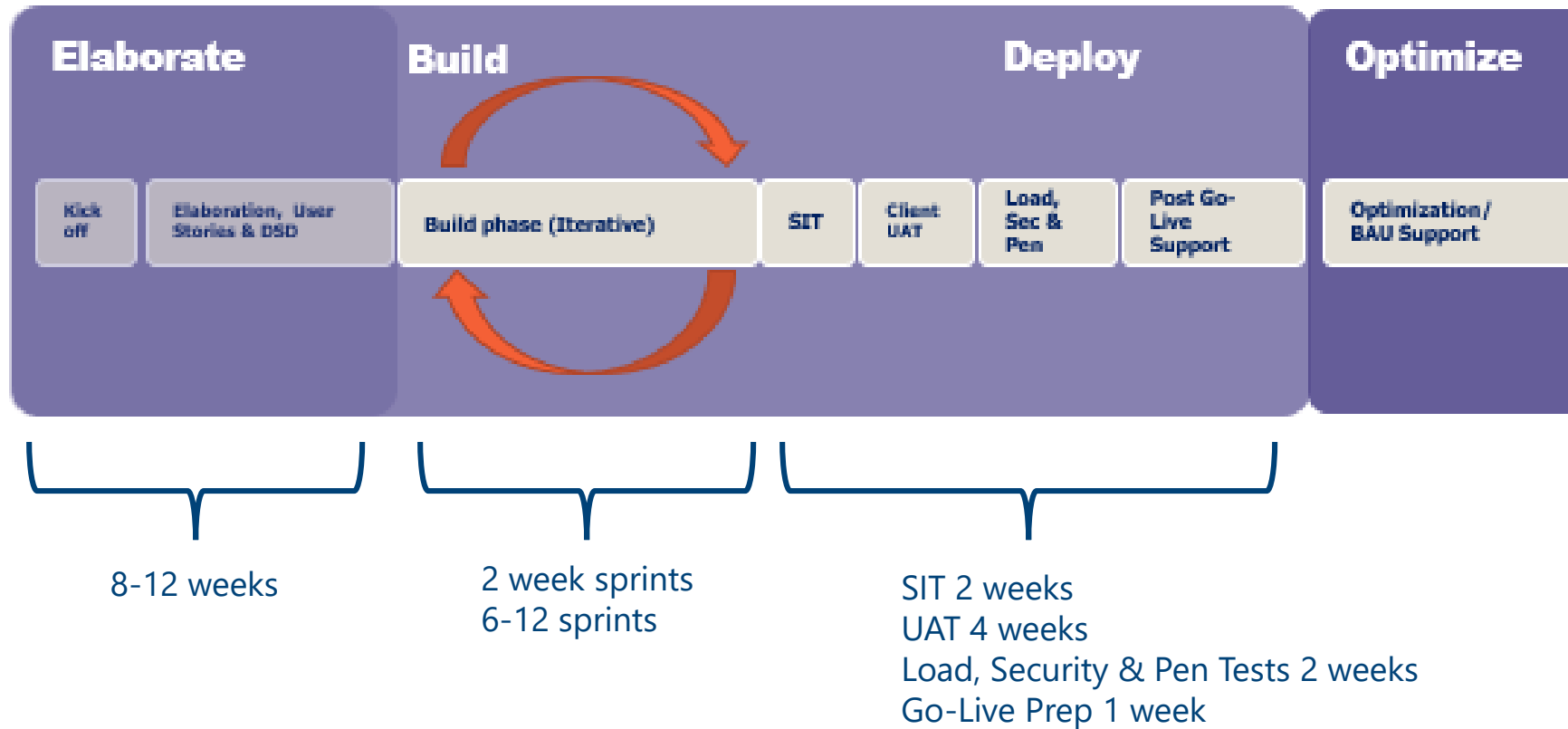
Avoka Delivery Model Project Planning & Execution



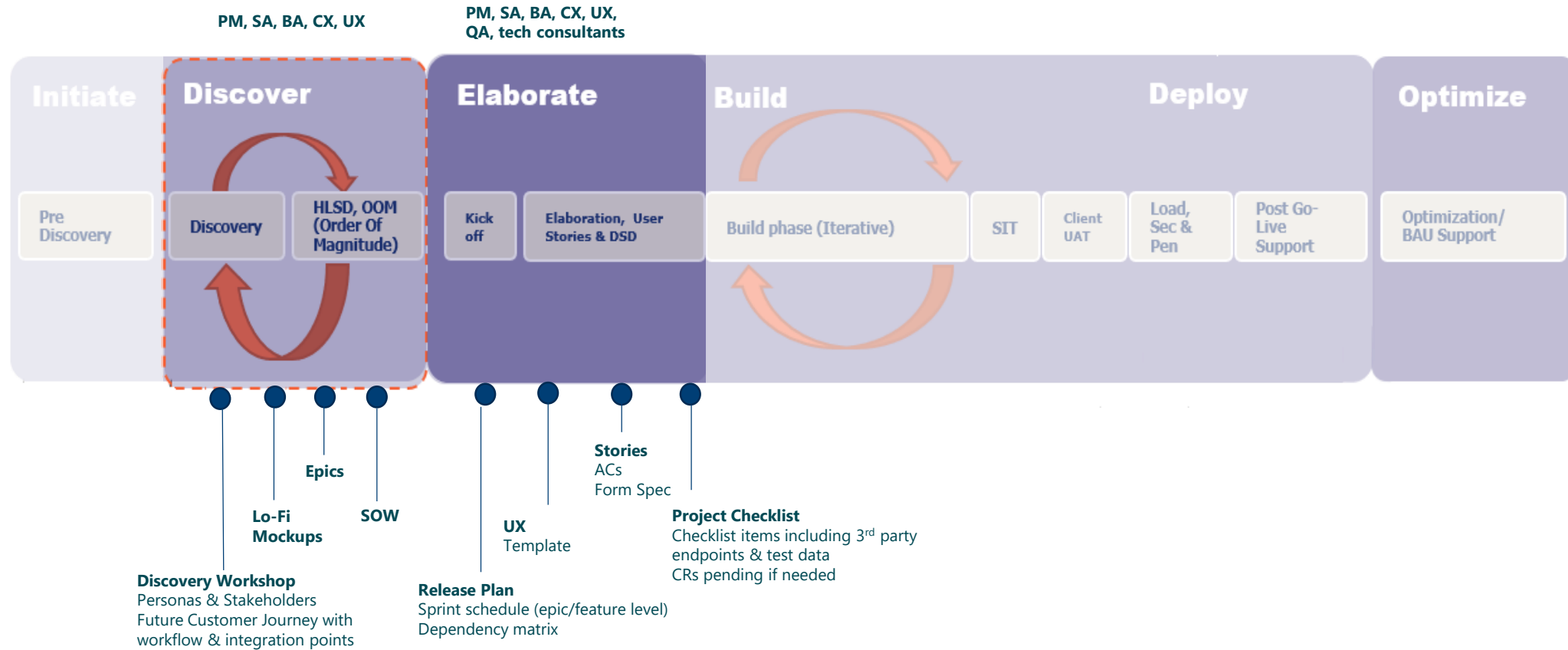
Avoka Delivery Model



Example Timeline



Discovery & Elaboration Phases



Avoka Client Services Team



Project Manager

- Has overall responsibility for Avoka Delivery
- Handles client stakeholder and management engagement
- Implements the Avoka delivery model with client
- Initiates and runs regular meetings, writes and sends regular reports
- Manages scope, schedule, and costs per the Statement of Work
- Manage the risks and issues process



Solution Architect

- Defines Avoka architecture for project scope
- Explains and advises the client on the architecture and how requirements are to be translated into project features and functionality
- Provides support for the Avoka delivery team
- Engages client with an enablement process/plan for client adoption and knowledge transfer



Business Analyst

- Serves as the Avoka gatekeeper for requirements management
- Assists client Product Owner and Business Analysts to manage and prioritize business requirements (flow diagrams, user stories)
- Provides bridge between client Product Owner and Avoka delivery team
- Manages change request receipt and estimation process
- Contributes to triage process for defect/issue resolution



Technical & QA Consultants

- Responsible for the technical delivery/quality assurance of specific stories as part of the overall team delivery of the Avoka solution
- Support the progress of Avoka solution through the client's test stages to live

| Additional Avoka Team Members



Account Manager

- Owns the account relationship and contractual communication
- Works with client to ensure completion of 3rd party contracts
- Manages roadmap for client relationship after Avoka Client Services team has completed project



CX Consultant

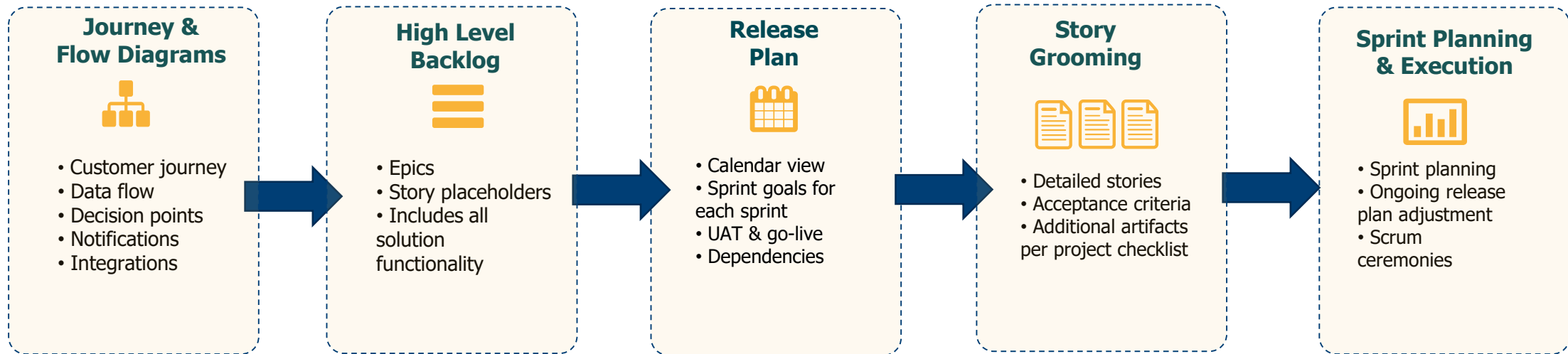
- Engage the client on customer experience and hold design workshops
- Review customer experience with delivery team
- Review development work prior to delivery to ensure customer experience expectations are met
- Provide input and guidance on optimization recommendations based on analytics data



UX Consultant

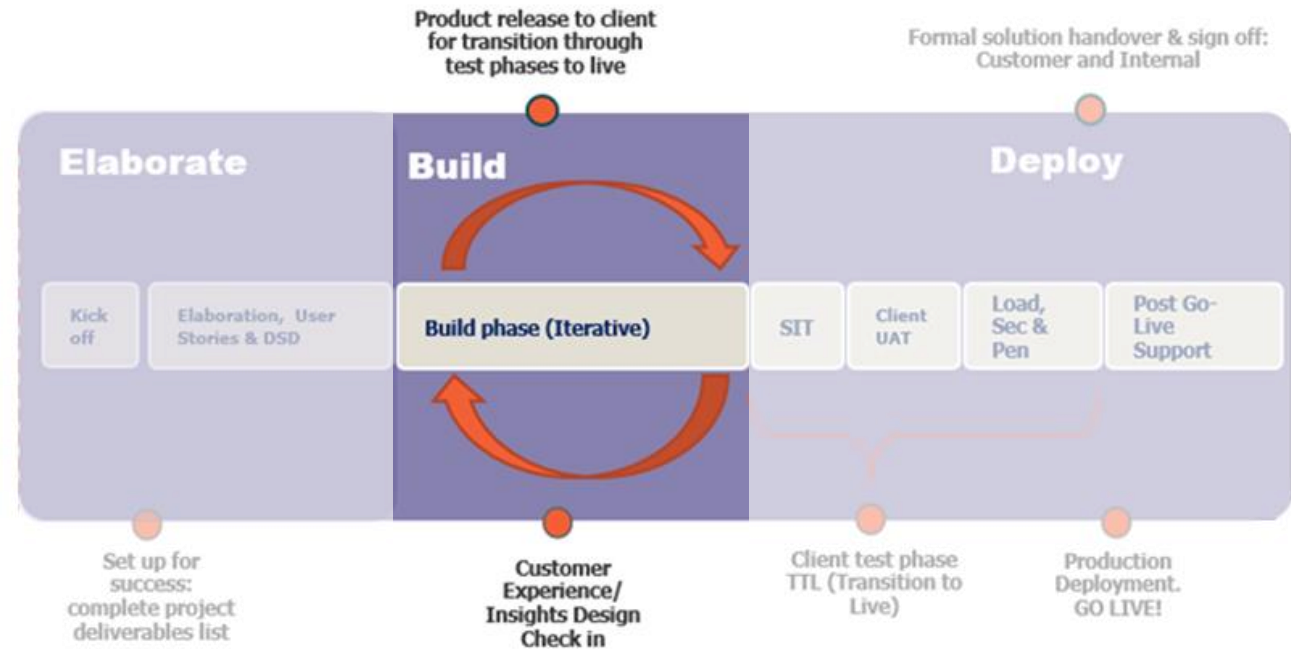
- Ensures that solution complies with client brand standards and style guidelines
- Works with clients to ensure that brand standards and assets are available
- Develops form templates

Discovery & Elaboration

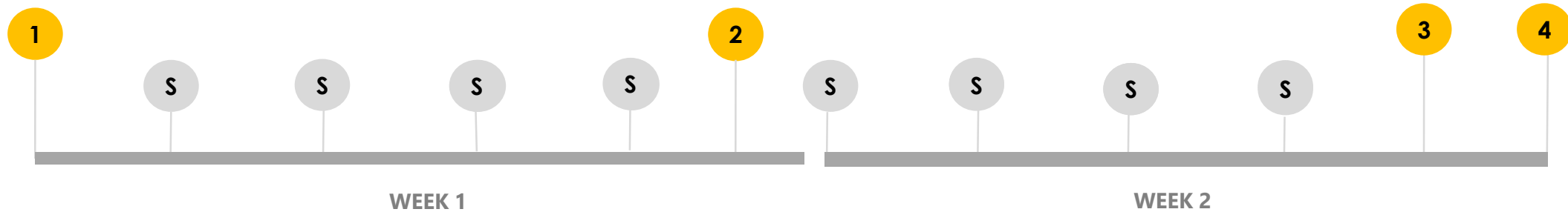


Build Phase

- Development & testing
- Sprint planning
- Daily standup
- Story grooming
- Sprint demo
- Sprint retrospective



Scrum Ceremonies



1. SPRINT PLANNING

Scrum team reviews Release Backlog user stories & associated acceptance criteria as a group. Developers can ask any questions of the Product Owner and Business Analyst ensure that all requirements are fully understood.

After the initial questions, the development team uses a process called “Planning Poker” to identify a point estimate for each user story. This is a relative estimate which uses past work for the team as a baseline. Teams commit in this meeting to the scope of a sprint.

S. DAILY STAND-UP

In this brief daily meeting, scrum team members are tasked with sharing three key pieces of information: what tasks did you work on yesterday, what tasks do you plan to work on today, and are there any items blocking you from performing these tasks. Often blockers can be removed by other team members who didn’t know they were creating a blocker!

These meetings are intentionally brief. They are moderated by a scrum-master who ensures that any extraneous conversations are tackled in break-out sessions.

2. STORY GROOMING

Product Owner and team review backlog stories and identify any gaps that need to be filled prior to sprint planning for the following sprint.

3. SPRINT DEMO

Product Owner and project stakeholders join the scrum team to view the progress on the current sprint’s work. Through this process, they get to see the completed stories. In addition, they receive an update if any stories have fallen out of (or into) scope for the sprint.

4. RETROSPECTIVE

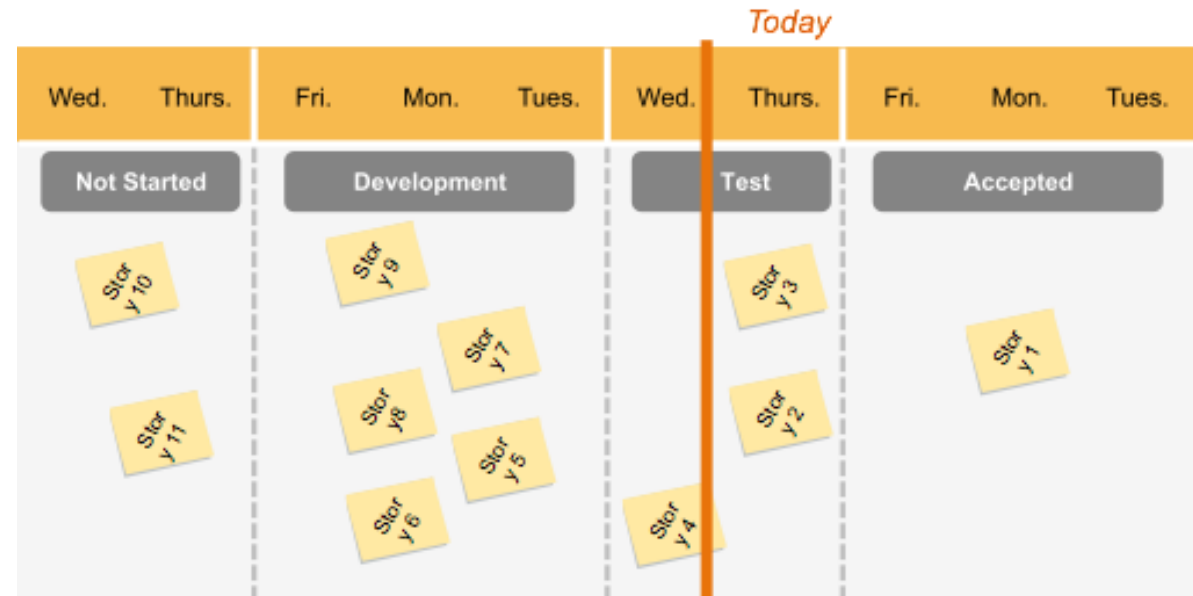
At the end of each sprint, all team members discuss the perceived successes and failures of the just-concluded sprint. Through this process, key bottlenecks and process challenges are identified.

The overall goal of the retrospective is to ensure that each future sprint is superior to the current sprint through open communication and iterative improvement.

Definition of Done

Defines the team's common understanding of the criteria that need to be met before a story can be completed.

- Acceptance criteria met
- Code in source control
- Unit/integration tests complete
- Code reviews complete & standards met
- UX/BA review complete (if applicable)
- Functional QA complete in test environment
- No Severity 1 or 2 defects
- Deployed to UAT for client acceptance



High-performing scrum teams limit work in progress to ensure work gets to done!

Sprint Workflow

Ready for Development	In Development	Ready for Review	Ready for Test Development	In Test	Ready for UAT Deployment	In UAT (Done)
<p>Items meet Definition of Ready</p> <p>During sprint planning, items have been committed to be completed by the team</p> <p>Critical or high priority defects blocking committed items can be added during sprint, but scope otherwise frozen after sprint planning</p>	<p>Items are under development</p> <p>Optional: separate Blocked column to indicate blocked items</p> <p>Assignee: developer</p>	<p>Items have been completed in development environment and meet Definition of Done (including tests)</p> <p>Merge request has been submitted for review</p> <p>Assignee: technical lead (for code reviews) or BA (for cosmetic/UX items)</p>	<p>Review has been completed and code is ready for deployment to test environment</p> <p>Test deployments are recommended daily</p> <p>JIRA resolution set on deploy</p> <p>Assignee: technical lead</p>	<p>Item has been deployed to test environment and Avoka QA is validating</p> <p>Critical or high defects that block acceptance should fail story back to Ready for Development or (optional) Test Failed column</p> <p>Low and medium priority defects logged to backlog</p> <p>Optional: Test Failed column</p> <p>Assignee: QA</p>	<p>Code has been accepted by QA and is ready for client to accept when deployed to UAT</p> <p>UAT deployments recommended throughout sprint</p> <p>Assignee: technical lead</p>	<p>Within a sprint, "done" means the item is ready for UAT, through a separate workflow and JIRA board</p> <p>Assignee: client product owner</p>

Release Planning

Releases

QUICK FILTERS: Released Unreleased Archived

Version name Start date (optional) Release date (optional) Description (optional)

Version	Status	Progress	Start date	Release date	Description
Release 4	UNRELEASED	No issues			Release 4
Release 3	UNRELEASED				Release 3
Release 2	UNRELEASED				Release 2
Release 1	UNRELEASED				Release 1

Version: Release 1
68 issues in total

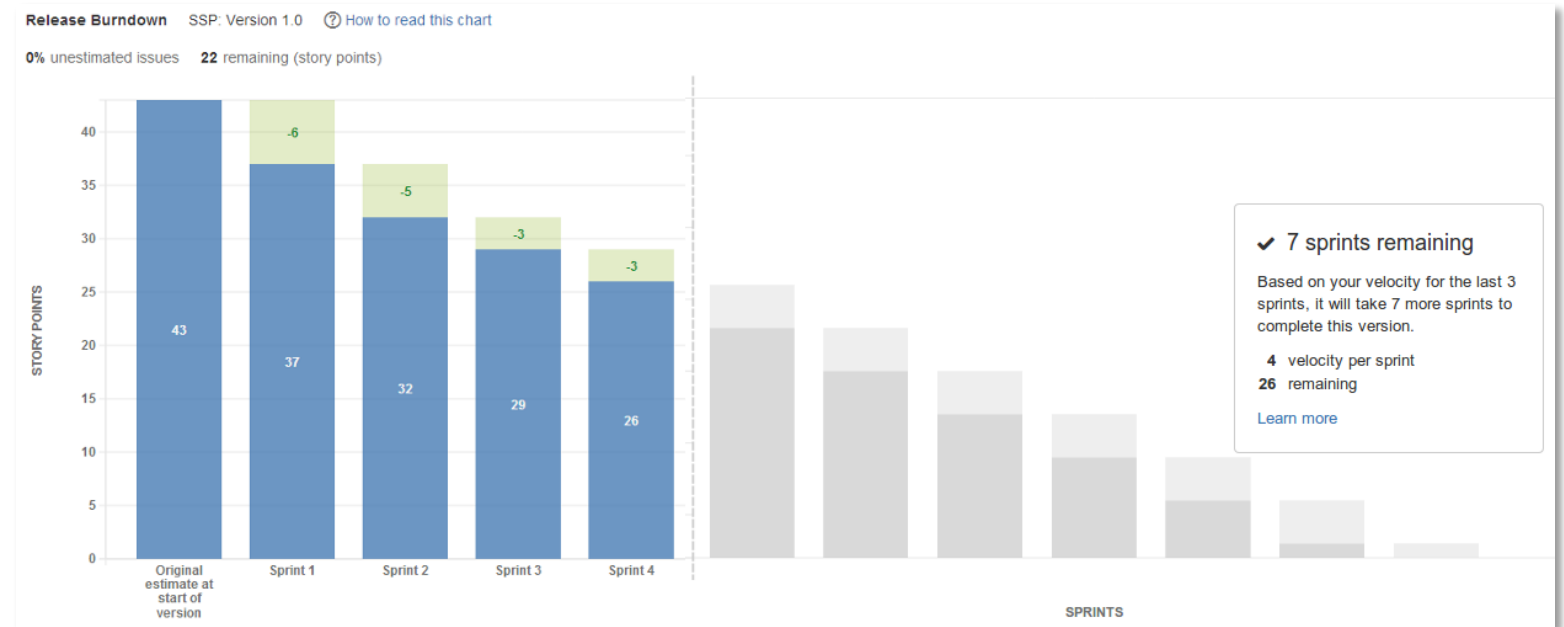
- Done 2
- In Progress 29
- To Do 37
- All issues

Management Software (v7.9.2#79002-sha1:3bb15b6)

A release is a logical grouping of functionality (e.g., MVP) that is planned ahead of time and may or may not be a public launch

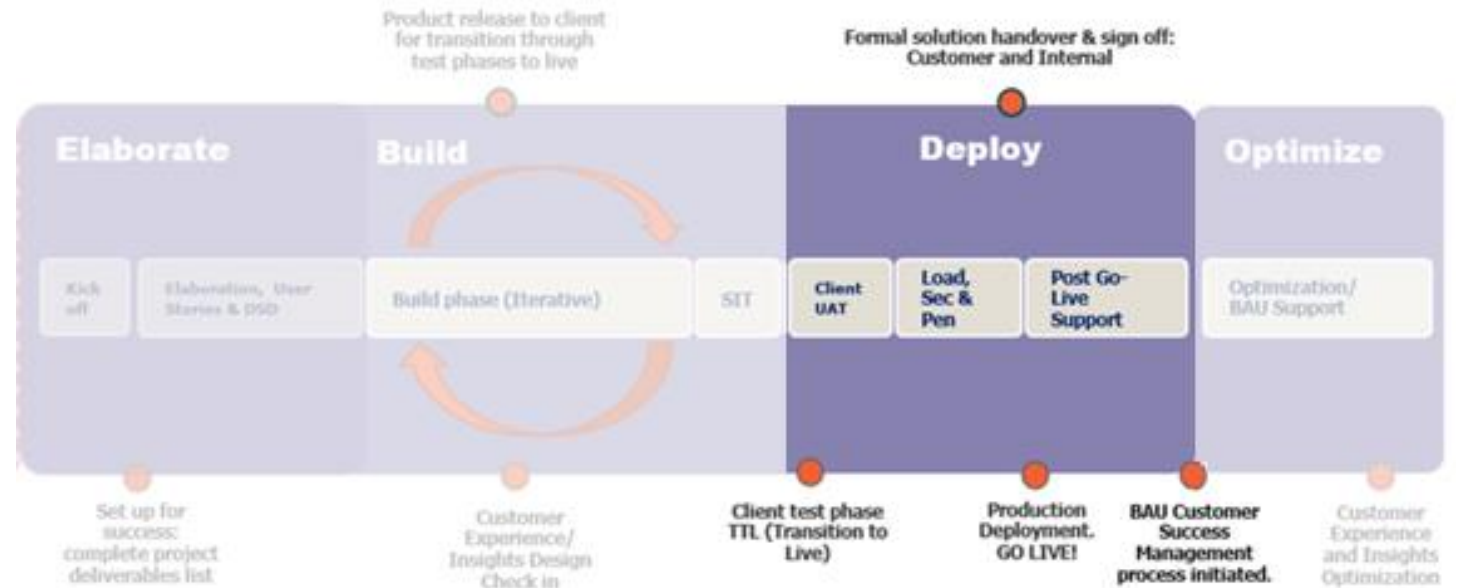
Release Burndown

- Shows the team's progress towards release objectives
- Release = defined scope that takes multiple sprints to implement
- Requires up front planning to identify and size scope for a release
- Can be an effective way to assess overall project health



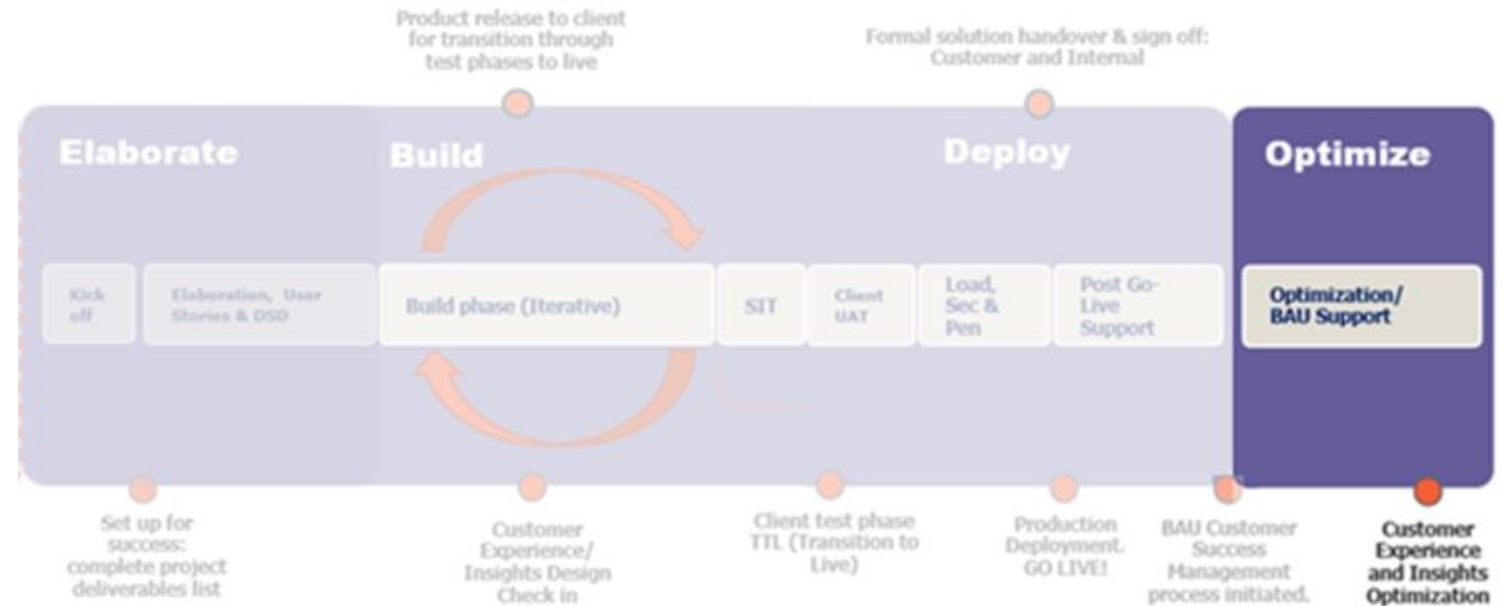
Deploy Phase

- Deployment to production
- 4-8 weeks
- Final client UAT after build complete
- Load, security & penetration testing by client
- Production deployment & end-to-end prod testing by client
- Avoka support handoff meeting



| Optimize Phase

- Avoka platform support
- Ongoing optimization of solution based on production analytics (Insights) data
- Platform upgrades (recommended 1x year)
- 3rd party upgrades (as needed)



Thank You

temenos.com