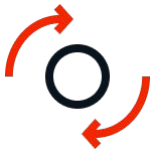
outsystems

Project Delivery Playbook

June, 2023

Using Agile Methodology

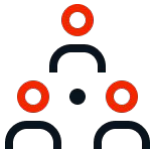
Timebox Principle



Iterative Development

Transparent Project Management

Flexibility

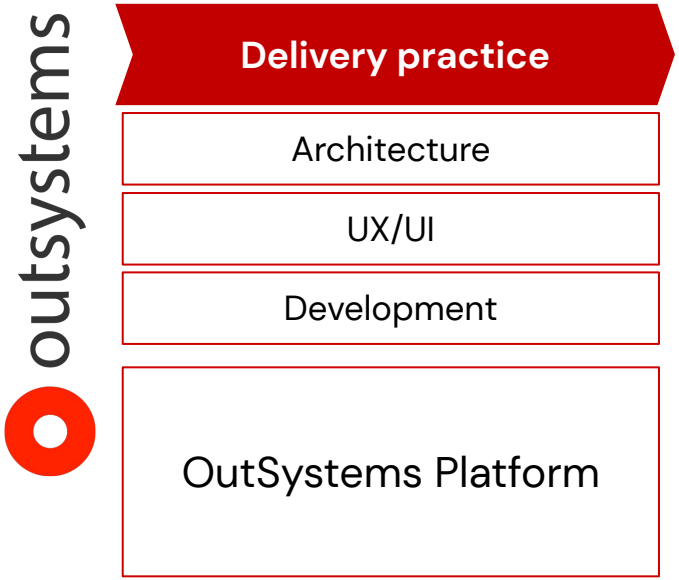


Collaborative Approach

Room for Feedback

Full Visibility on Execution

Reaching the full potential



Keeping **High Speed**

delivering **Maximum Value**

with **Quality**

Skill sets mapping

Biz Innovation	Strategic, innovative ideas to digitize business	<Customer>
Biz Analysis	Understand the requirements, design and validate the digital system / application	OutSystems
Custom Software Development	Develop and deliver the application	OutSystems
Systems Integration & Architecture	Understand the complete IT picture, integrate with existing systems, define the enterprise architecture	<Customer>

<Customer>

Development capacity from <customer> (when applicable)

OutSystems Team Responsibilities

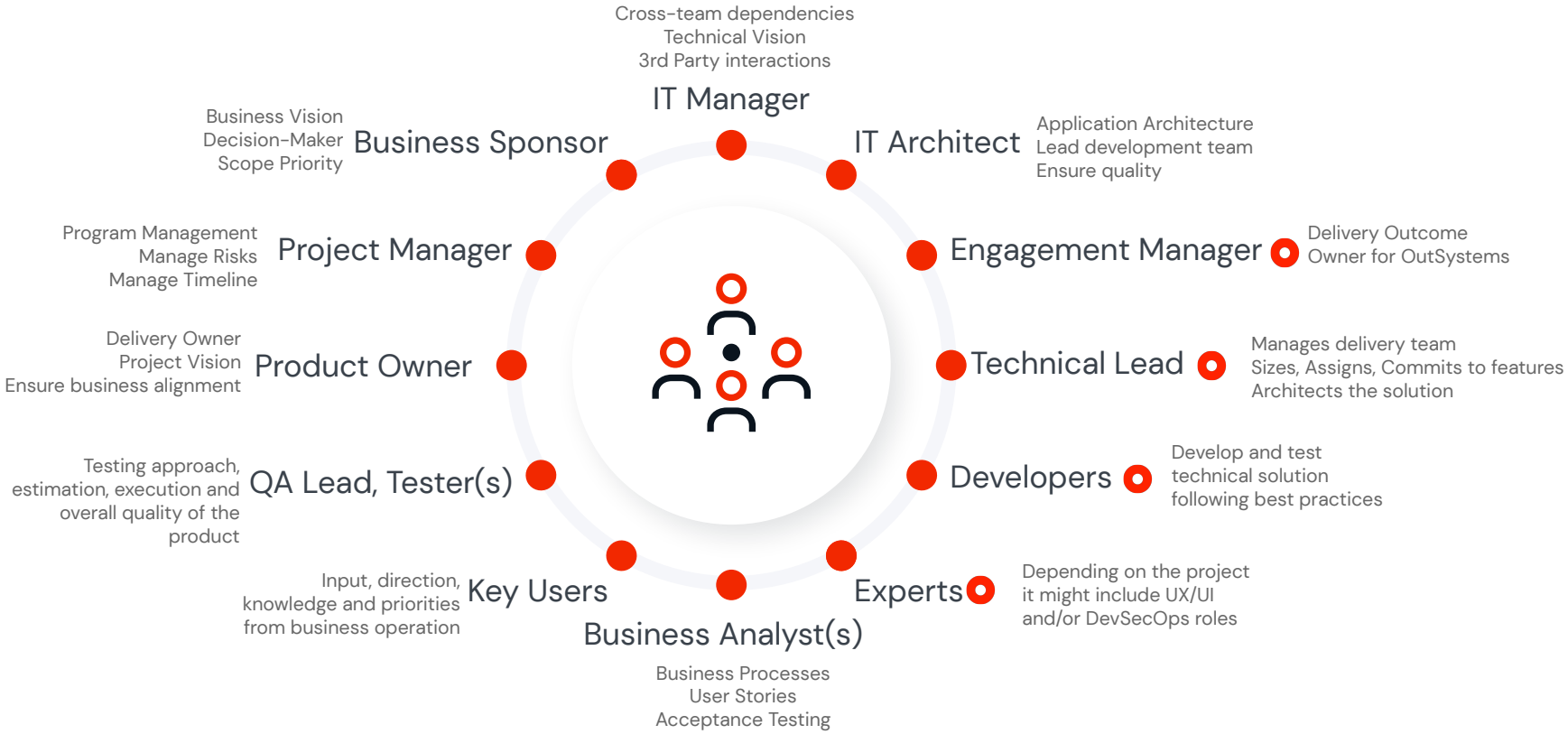
OutSystems	
Program Manager	. Ensures the several project tracks are well **orchestrated** towards meeting **program strategic goals**
Engagement Manager	. Responsible for the OS **delivery services execution** . Acquires **business context and knowledge** . Facilitates the creation of a **project vision** and ensures alignment on the implementation . Helps **validating the User Stories** and Acceptance Criteria in alignment with the business, promoting convergence and simple, pragmatic solutions (when 100% allocation) . **Coaches the delivery process** . **Manages the backlog** in tight cooperation with the customer's PO . **Represents the business** before the delivery team . **Manages engagement topics** (change requests, status reports, risks, budget, and resources) . Conducts **functional testing** (when 100% allocation, it includes end-to-end tests) . Conducts feature delivery **demos** and **collect feedback** . Continuous monitoring and improvement, through **sprint retrospectives** . Ensures proper **project handover** (when 100% allocation)
Technical Lead	. Designs, sizes and builds a functional and technical **system architecture** based on the business needs and drivers . Designs and sizes third-party integrations, being the **technical point of contact** . **Leads the delivery team**, ensuring teamwork, commitment, and control of the progress . Is the **responsible for the delivery** on each sprint and at the end of the release, of a viable and scalable solution . Is the responsible for solution non-functional requirements (like solution performance, scalability, and security) . **Guarantees quality** procedures and OutSystems behaviors . Supports feature delivery **demos** . Manages solution **staging** life-cycle, conducting QA and production deployments and supporting overall solution rollout procedures . Capacity reserved for development activities: 30-50% of DM allocation for teams with 1 Dev, 0% with 2 or more
Developer	. **Develops and tests the technical solution** following the engineering best practices
Expert (UX, Architect, PlatformOps, Front-End,...)	. **Expert** executing a service in the context of the project. Can be a UX, UI, Architecture or Platform expert fully aligned with the project goals

<Customer> Team Responsibilities

<Customer>	Project / Program Manager	. Embraces the **Program/Project Management** activities at the customer side. Manages and mitigates risks. Manages timeline
	Business Sponsor	. Provides **business vision**, direction and guidance . Participates in **feature negotiation**, **demos** and **steering meetings** . Is a **decision making** authority . **Scopes prioritization**, considering both strategic and business operation goals
	IT Manager / Architect	. Coordinates **cross-team dependencies**, the support from the rest of the IT organization (including infrastructure changes, availability of test environments and release process), the delivery of cross-product software dependencies and the removal of cross-product impediments within the chain . Leads the **technical vision** and direction for the product . Manages the **integrations / data migration** . Manages the **3rd party interactions**
	Product Owner	. **Manages the Business Analyst** output . Manages the **product vision** and overall output . Is the delegated **decision making** authority (*) . Monitors the **cost-benefit trade-offs** in the project . Represents the end users, **creates User Stories** . Facilitates the product **backlog prioritization** and **sprint planning** . Performs the **acceptance testing** . Performs the **demos** to business . A senior BA sometimes fulfills this role
	Business Analyst	. Promotes the complete definition of the **business processes** next to key-users . Conducts **Business Analysis**, clarifies processes, breaks them down into User Stories and provides their Acceptance Criteria . Perform **acceptance testing**
	Key User	. Provides **input**, direction and **priorities** from the **business operations**
	Tests / QA coordination	. Define **test cases** based on US acceptance criteria. Coordinate User Acceptance Testing (UAT) and **perform tests**

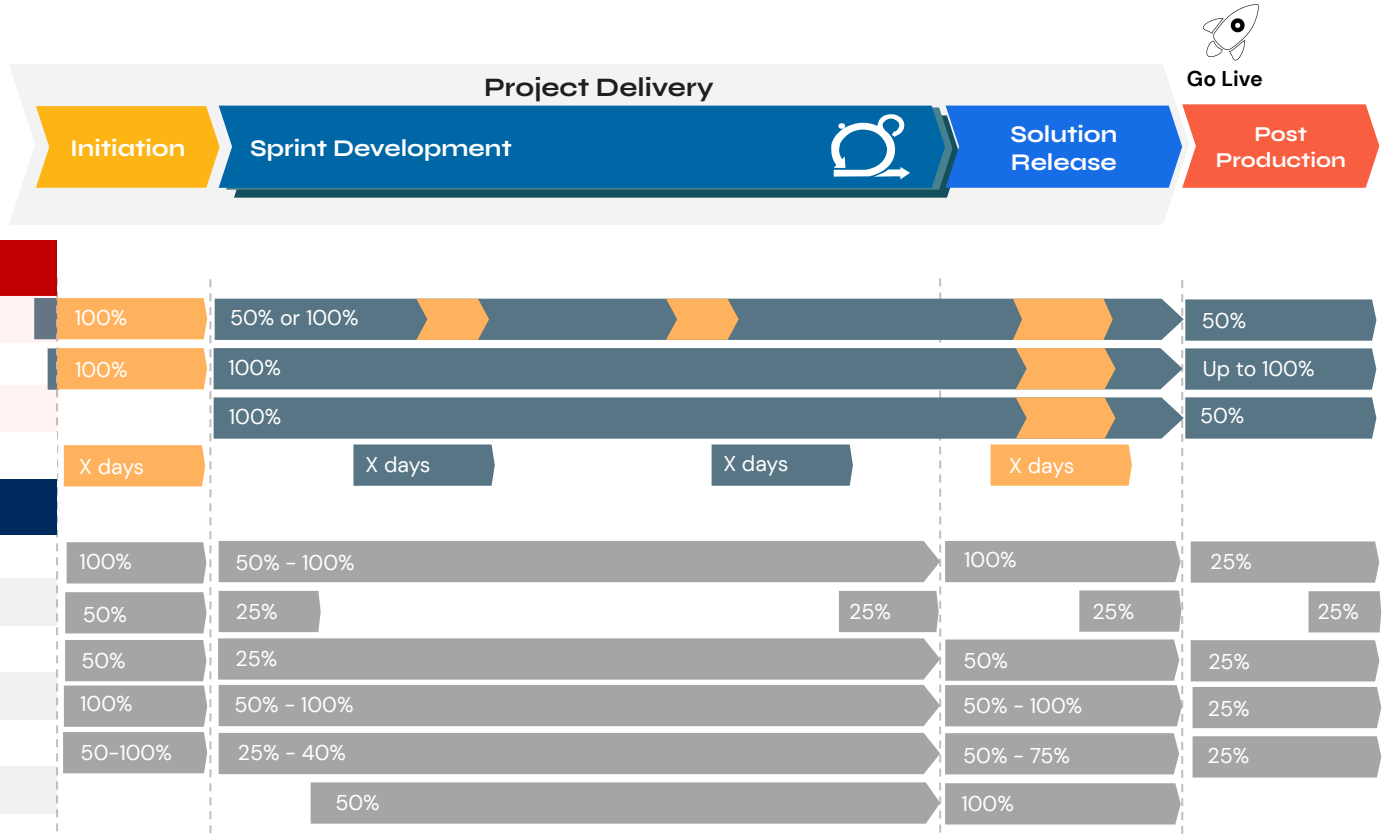
(*) Product Owner should have sufficient decision making authority to govern the entire Project. If features or data are dependent on another Team, this might cause delays.

Project Responsibilities



OutSystems projects are heavily people-centric and collaborative

Reference Allocation



(*) Customer availability is presented as reference

outsystems

Project Phases

Go Live

Project Delivery

Initiation | Sprint Development | Solution Release | Post Production

Initiation

Where we start the actual project. The key focus is on understanding the business and users needs to build with <customer> the vision of the future application:
How will it answer to the most important Users Stories? What will it look like? What will its architecture be?

Sprint Development

An incremental and iterative process delivers the application. This means that every 2 to 3 weeks there is a checkpoint with the users and stakeholders to demo the part of your application developed during that time. These checkpoints are also the way to respond to business changes.

Detailed analysis, development, and testing are done incrementally, so if a change or a new requirement that brings significant value to the business comes up, it can replace another one with lower business value.

During the Sprint Development and after each demo, the users are also invited to test the application. Doing so allows them to provide feedback that will improve the application's overall quality and usability.

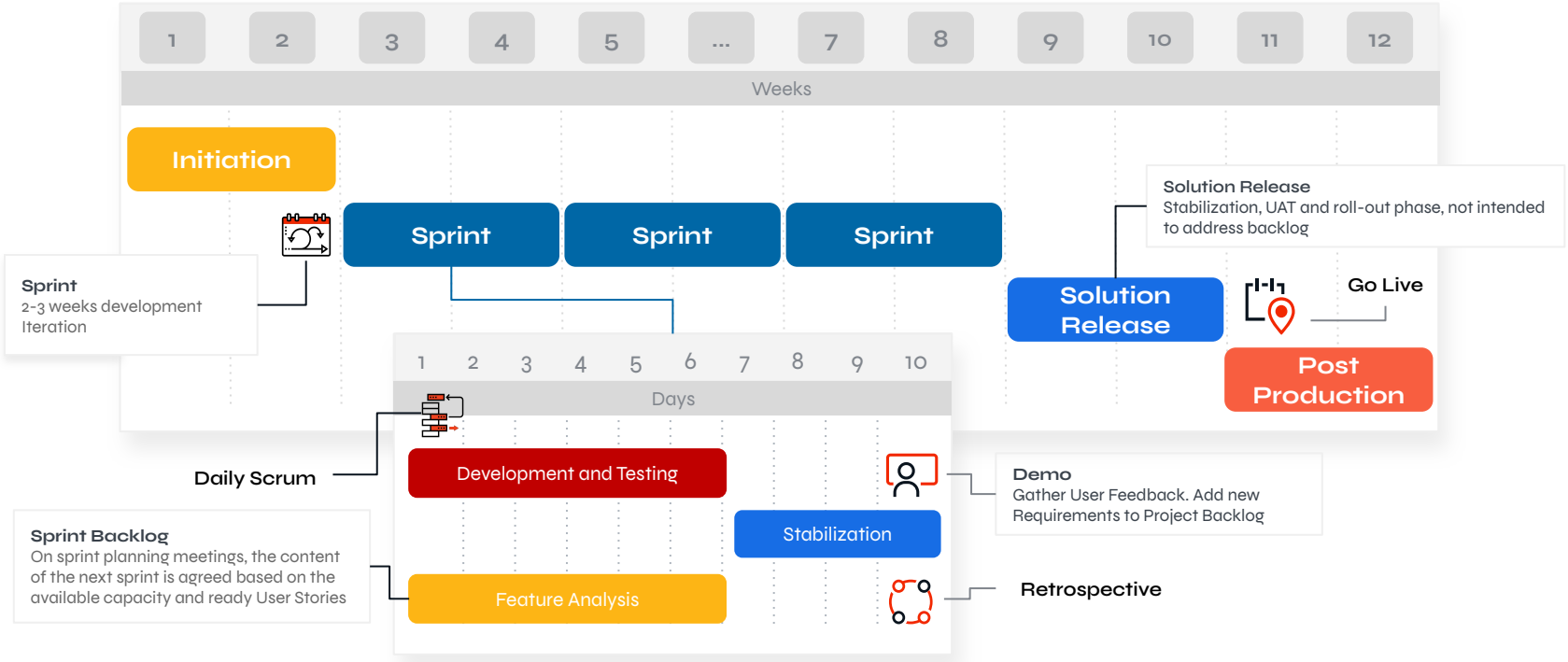
Solution Release

This phase is the moment for the stakeholders and users to test the entire application end to end thoroughly, and for the delivery team to make the final improvements before the new solution goes live. At the end of this stage, the new solution is live, and the business starts benefiting.

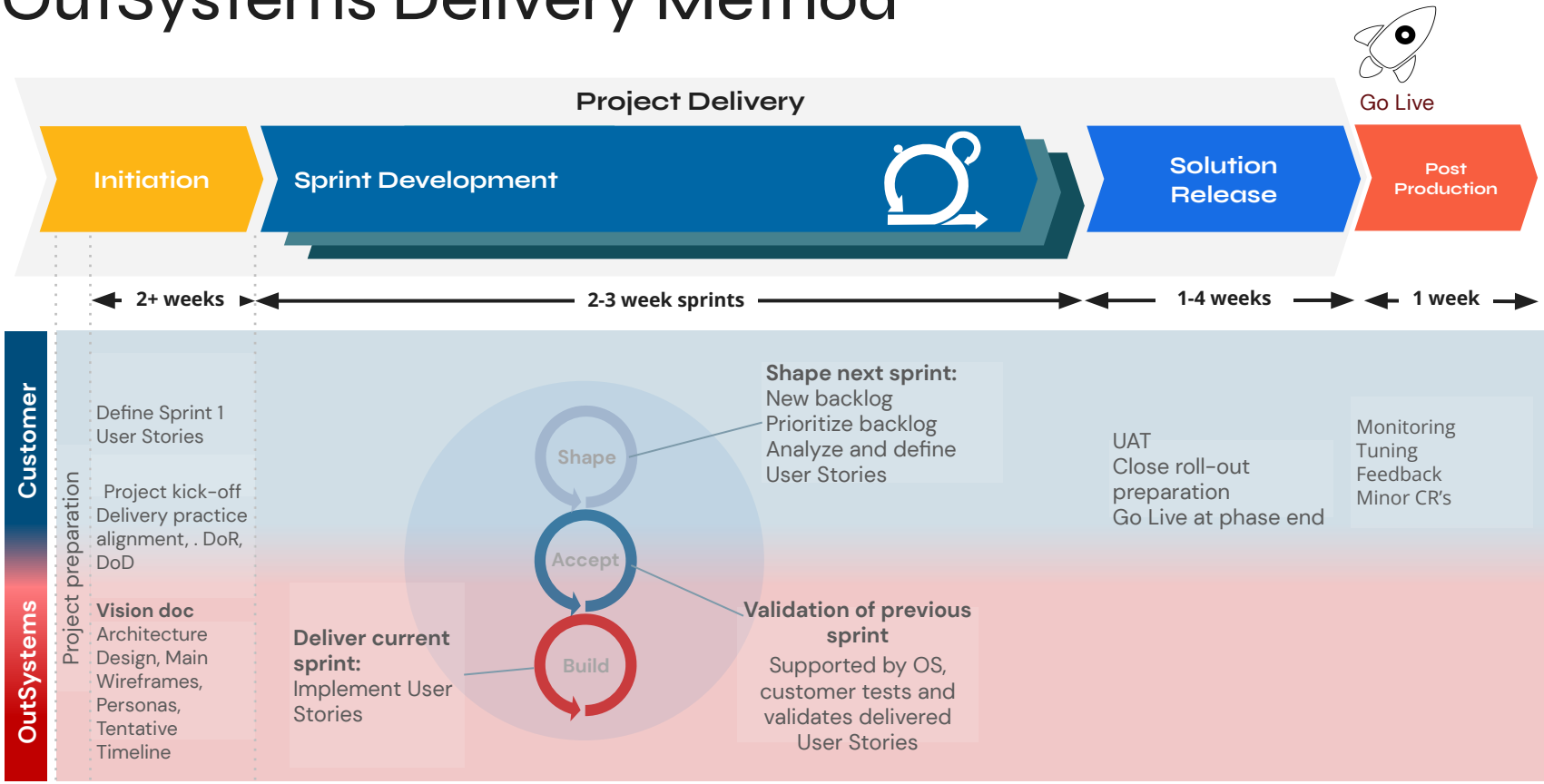
Post Production

When users start using the application, some usability and performance issues may arise. This phase aims at performing a fine-tuning that improves the adoption and promotes an excellent user experience. This is also the right moment to plan for the future and discuss how to ensure the application continuously supports the business over time.

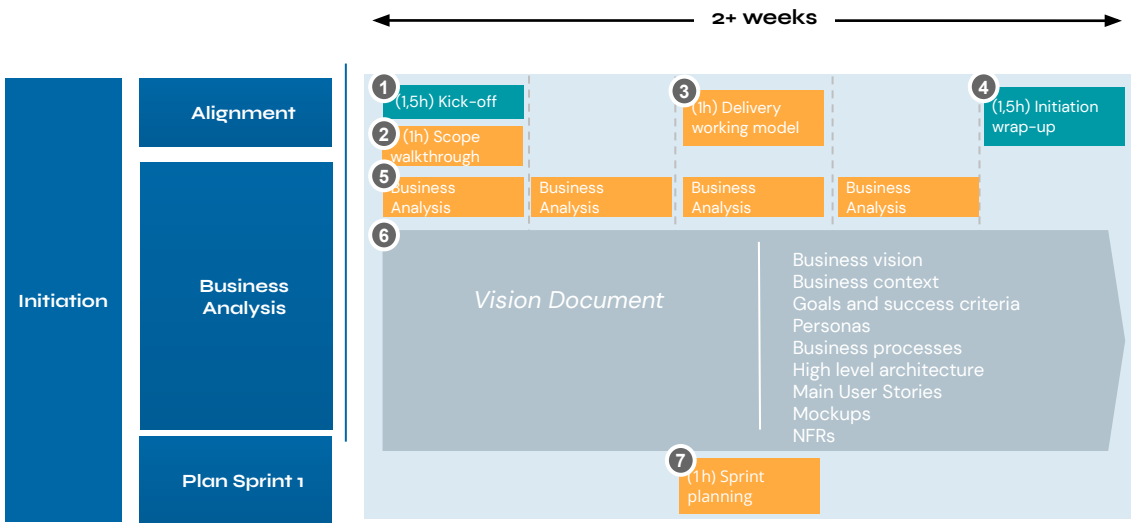
Agile Delivery with OutSystems



OutSystems Delivery Method



Initiation

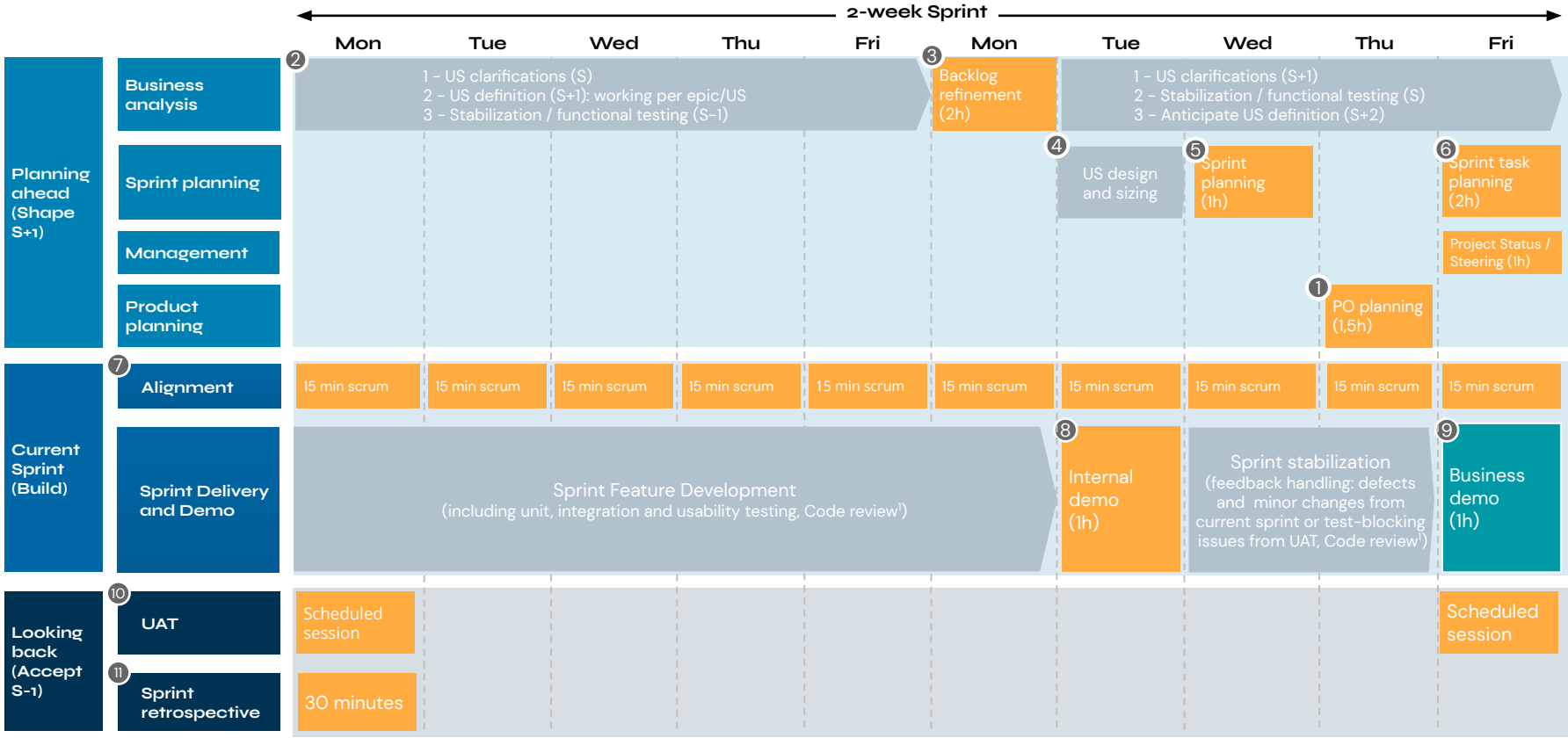


* For longer duration releases (by reference > 3 months of Sprint Development), the Initiation should have another week: the first focusing on gathering info on site with the customer, and the second week to compile and close phase deliverables (a week wrap-up session should still take place at the end of week 1)

Initiation - Events Calendar

Event	Inputs	Activities	Outputs	Participating
1 Kick-off	▪ Kick-off presentation	▪ Kick-off meeting ▪ Align project goals, success criteria, plan ▪ Establish communication plan and schedule all regular sprint activities	▪ First alignment between all major stakeholders	▪ All (mandatory attendance from stakeholders, sponsors and key-users)
2 Scope / OS services walkthrough	▪ Proposal ▪ Sizing ▪ Sales handover session output	▪ The EM presents the included OS offer services and the sizing, validating important assumptions and discussing major issues / risks	▪ Align on how the project timebox / available capacity was determined ▪ Align on version 0 of the scope	▪ BS, PM, PO, BA ▪ OutSystems team (incl. Sales)
3 Delivery working model	▪ Project Delivery Playbook	▪ The EM presents the iterative working model, focusing on its main challenges on how, together, the full <customer>-OS team can overcome them ▪ Establish DoR, DoD and how these can be met	▪ Full team is clear on how sprints will work and their role in it	▪ PM, PO, BA, QA ▪ OutSystems team
4 Initiation wrap-up	▪ Initiation week achievements	▪ Present the achievements and conclusions of the initiation week ▪ Re-validate project plan	▪ Vision Document ▪ Project Plan ▪ Mock-ups	▪ All (mandatory attendance from stakeholders, sponsors and key-users)
5 Business analysis meetings	▪ Project scope ▪ Business context	▪ Define the business process diagrams ▪ Identify the personas ▪ Define the User Stories and have their validation from both business and IT	▪ Sprint backlog and plan	▪ QA, KU, PO, BA, IT
6 Vision Document	▪ Project Data ▪ User research data ▪ Business analysis	▪ Project Goals and business context ▪ Business processes and main user stories ▪ Mock-ups and information architecture	▪ Vision Document	▪ EM, TL
7 Sprint 1 Settlement	▪ User Stories	▪ Prioritize and estimate the backlog ▪ Decide what's going to be delivered in sprint 1	▪ Sprint 1 deliverables	▪ PM, PO, BA, QA ▪ OutSystems team

Sprint Development



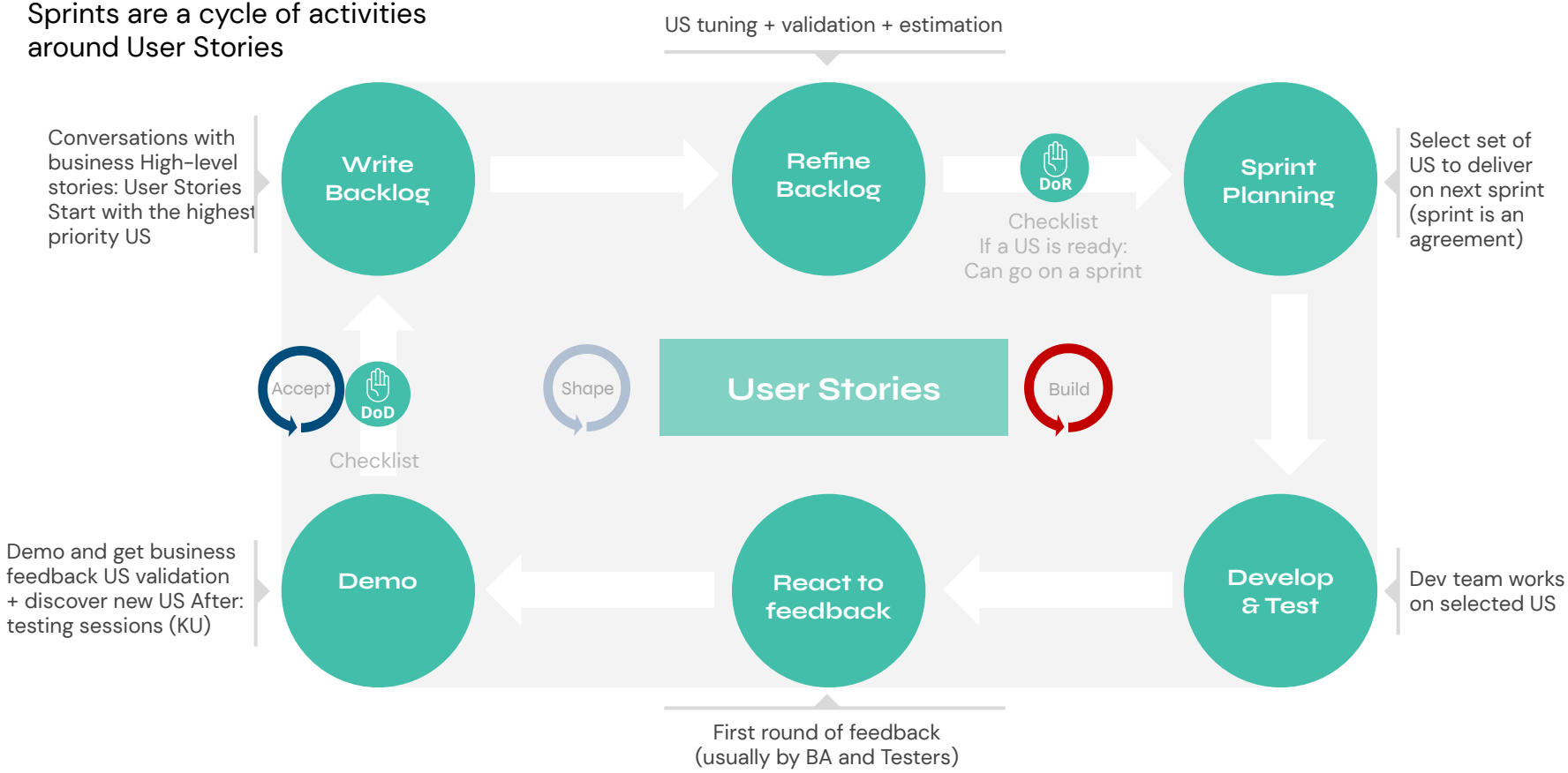
[1]as you develop always validate if you are applying the best practices and use them when not

Sprint Development - Events Calendar

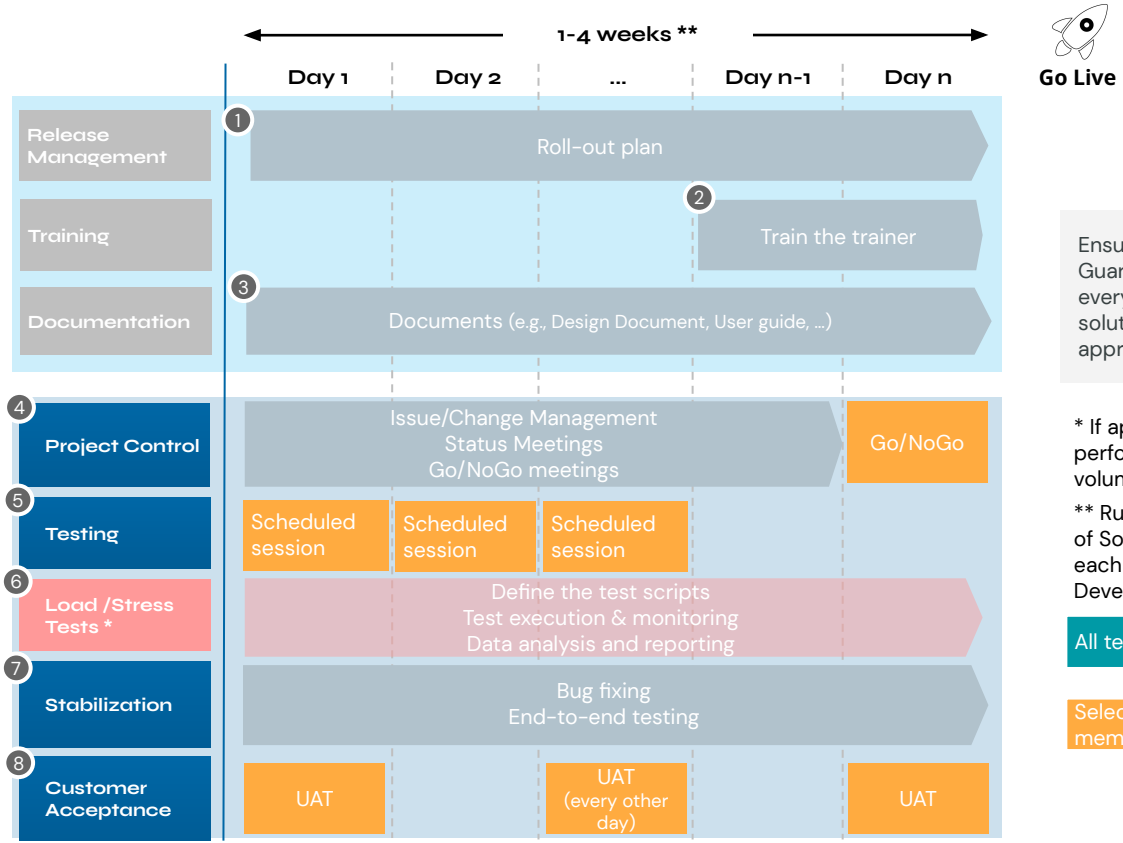
Event	Inputs	Activities	Outputs	Participating
1 Product owners planning	▪ Business findings ▪ Stakeholder input ▪ Feedback from previous sprint	▪ Plan the product roadmap and discuss the overall progress against the goals ▪ Identify the backlog epics (always mapped against the business process) ▪ Prioritize the backlog ▪ Identify the goal, candidate epics and required BA Squads for the next sprint	▪ Prioritized backlog ▪ Next sprint goal ▪ Candidate epics and Squads for the next sprint	▪ PO ▪ EM
2 User Story definition	▪ Candidate epics for next sprint	▪ The BA breaks down epics and define the US (mapping back to process) ▪ Business and IT dependencies are identified ▪ Close follow-up with the EM ensuring the US functional readiness	▪ Ready US candidates for the next sprint	▪ BA, IT, QA ▪ EM
3 Product backlog refinement	▪ "Functional DoR" US candidates for next sprint	▪ The BA presents the candidate US to Dev team ▪ Held in several sessions during week 1, as soon as a BA finishes	▪ Functional knowledge for the next sprint shared with the team	▪ BA, PO, IT ▪ OutSystems team
4 US design and sizing	▪ Get the US candidates ready for the next sprint	▪ Technical design for adding the US to the existing solution while meeting non-functional requirements ▪ US sizing	▪ Ready US candidates for the next sprint	▪ BA, IT ▪ OutSystems team
5 Sprint planning	▪ Get the US candidates ready for the next sprint	▪ The US design and sizing is presented by the EM, and a sprint scope settlement is agreed ▪ BA / IT do a final validation on design (adding to iterations during design step) ▪ This understanding is also used for distinguishing defects from changes	▪ Sprint backlog and plan	▪ BA, IT,PO ▪ OutSystems team
6 Sprint task planning	▪ Sprint backlog and plan	▪ Break prioritized US for upcoming sprint into tasks ▪ Define the work and delivery plan based on the task dependencies	▪ Sprint work and delivery plan	▪ OutSystems team
7 Sprint alignment	▪ Sprint progress	▪ Each person gives a short summary on the 3 questions: *"What I did yesterday / What I will do today / What impediments I have"* ▪ For efficiency, consider scrum of scrums when, otherwise, there would be more than 8 people involved (rule of thumb)	▪ Updated sprint progress ▪ Updated list of impediments and related mitigation actions	▪ BA, IT, QA ▪ EM, TL
8 Internal demo	▪ First sprint delivery version, after Dev team implementation and tests	▪ The Dev team presents the sprint to the BA and the QA, which can perform tests to support sprint stabilization until the business demo	▪ Sprint version ready for stabilization phase	▪ BA, QA, PO, IT ▪ OutSystems team
9 Business demo	▪ Potentially shippable increment of product ▪ Metrics from the sprint ▪ Updated release plan	▪ The BA and the EM present sprint to business: demo script takes the user's perspective to provide a walk through (telling a realistic story about) the end-to-end business process set as sprint goal	▪ Signed off US ▪ Improvements ▪ New stories	▪ All (mandatory attendance from stakeholders, sponsors and key-users)
10 UAT	▪ Sprint delivery	▪ BA, QA and KU execute the test cases to validate the US acceptance criteria ▪ Support by dev team: removing test roadblocks and addressing findings preventing acceptance	▪ Accepted US ▪ Feedback (defects, changes) ▪ Usability tests	▪ BA, QA, KU ▪ OutSystems team
11 Sprint retrospective	▪ Sprint experiences from last sprint	▪ Each person gives a short summary of the 3 questions: *"What went well / What is not going well / What can we do to improve"* ▪ For efficiency, there might be more than one retrospective with different attendees, to focus on different areas (ex one for tech and another for PM)	▪ 1 or 2 actions for improvement ▪ Own reflections	▪ All

Sprint Development

Sprints are a cycle of activities around User Stories



Solution Release



Ensure business value
Guarantee that everyone is using the solution appropriately.

* If applicable, based on performance or user volume requirements

** Rule of thumb: 1 week of Solution Release for each 5 of Sprint Development

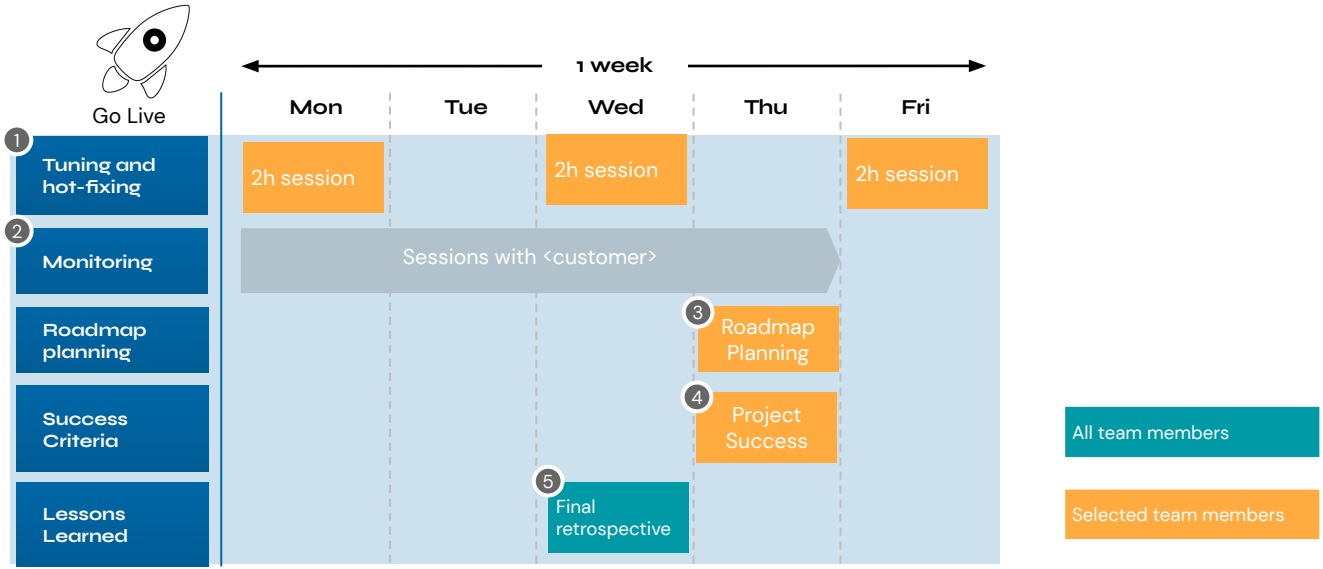
All team members

Selected team members

Solution Release - Events Calendar

Event	Inputs	Activities	Outputs	Participating
1 Roll-out plan	▪ Application Solution ▪ Deployment & bootstrap steps ▪ App replacement steps	▪ Step by step deployment script for a seamless installation ▪ Configurations and bootstraps, infrastructure, 3rd party integration readiness: all considering full architecture ▪ Plan to switch from the former app to the new one ▪ Understand the business impact of the solution and ensure the preparation of individuals, teams, and organization for the new processes and tools	▪ Roll-out plan	▪ IT, BA ▪ EM, TL
2 Train the trainer	▪ The app ▪ Main user stories	▪ Sessions with the key users/business champion to go through the main user stories	▪ Session video recordings	▪ KU, QA, BA, PO ▪ EM
3 Documentation	▪ The app ▪ OutDoc ▪ <Customer> IT requirements	▪ Deploy OutDoc on the environment ▪ Define with <customer> which are the documents needed (may imply additional effort not considered in sizing)	▪ OutDoc	▪ BA, PO, IT ▪ OutSystems team
4 Project Control	▪ Issues found in the QA procedures ▪ Change requests	▪ Backlog prioritization ▪ Status meetings to ensure progress ▪ Go/NoGo meetings with project stakeholders	▪ Decision towards the go live ▪ Prioritized backlog	▪ BA, IT, BS, PO ▪ EM, PM
5 Testing	▪ All user stories DONE ▪ Acceptance criteria ▪ End-to-End Test cases	▪ Execute the tests defined by QA, BA and EM ▪ Business process validation ▪ End-to-end testing	▪ Executed tests	▪ QA, KU, PO, BA, IT
6 Load/Stress Tests	▪ The app ▪ Infrastructure ▪ Main user stories	▪ Build the test scripts ▪ Execute multiple scenarios ▪ Data analysis and creation of a report	▪ Test scripts ▪ Report	▪ OutSystems Experts team
7 Stabilization	▪ Tests feedback ▪ Prioritized & estimated Issue list	▪ Hot-fixes to correct the issues ▪ Apply hotfixes to all environments ▪ Dev team addresses remaining code review issues	▪ The app ▪ Stable and scalable application, following OS best practices	▪ PO, BA, QA ▪ OutSystems Team
8 Customer Acceptance	▪ The app ▪ User stories	▪ Go through the user stories and validate the functionality against the business needs ▪ Accept (or reject) delivered functionality	▪ Delivery sign-off ▪ Go to production	▪ BA, QA, IT, PO, BS ▪ EM, PM

Tuning


Go Live
1 week
Mon
Tue
Wed
Thu
Fri
1
Tuning and hot-fixing
2h session
2h session
2h session
2
Monitoring
Sessions with <customer>
3
Roadmap planning
Roadmap Planning
4
Success Criteria
Project Success
5
Lessons Learned
Final retrospective
All team members
Selected team members

Post Production - Events Calendar

Event	Inputs	Activities	Outputs	Participating
1 Tuning and hot-fixing	▪ Identified quick-wins ▪ Feedback from the Key Users	▪ Develop and deploy the application improvements	▪ The app	▪ BA PO ▪ EM, DM
2 Monitoring	▪ ServiceCenter logs ▪ Performance Monitor ▪ Field observation	▪ Look at signs of slowness (slow queries, actions, timers or screens) ▪ Check for errors	▪ Feedback ▪ Backlog	▪ PO, BA, IT ▪ EM, TL
3 Roadmap planning	▪ Application backlog ▪ Identified opportunities	▪ Discuss the next steps ▪ <Customer> autonomy	▪ A plan (eventually)	▪ BS, PO, IT, BA ▪ EM, PM
4 Success criteria	▪ Project Charter ▪ Kick-off presentation	▪ Go over the results and assess the achievements ▪ Check the objective metrics, like financial optimizations, FTE reduction or better response times	▪ Case study (eventually)	▪ BS, PO, IT, BA ▪ EM, PM
5 Final Retrospective		▪ Conduct retrospective covering the business drivers/success criteria, challenges, sales/planned/actual, margin, NPS/CSAT results and lessons learned ▪ Include on the retrospective your manager, project team, involved salesperson and RSM	▪ Delivery sign-off ▪ Go to production	▪ <Customer> team ▪ OutSystems team

outsystems

Appendix

Glossary

CR	Change Request
CSAT	Survey sent to the customer to assess the customer satisfaction of the service delivered.
DoD	Definition of Done. It is a comprehensive checklist of necessary, value-added activities that assert the quality of a feature and not the functionality of that feature
DoR	Definition of Ready. A “ready” item must be clear, feasible and testable. Set minimum criteria that you must meet before it is ready to include in a sprint.
Epic	Epics are significant features that encompass many user stories that you deliver almost always over a set of sprints.
NFR	Non-functional requirement specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They contrast with functional requirements that define particular behavior or functions. Examples: robustness, quality, accessibility, compliance, reusability, ...
NPS	The Net Promoter Score is a customer loyalty metric developed with the objective to determine a clear and easily interpretable customer satisfaction score which can be compared over time or between different industries. The NPS assesses to what extent a respondent would recommend a particular company, product or service to his friends, relatives or colleagues.
Sprint	A time period (typically 2–3 weeks) in which development occurs on a set of backlog items that the team has committed to — commonly referred to as a time-box or iteration.
UAT	User Acceptance Tests consist of a process of verifying that a solution works for the user. It is not system testing (ensuring the software does not crash and meets the requirements documented) but rather ensures that the solution works for the user (i.e., tests whether the user accepts the solution).
US	User stories are short, simple description of a feature told from the perspective of the person who wants the new capability, usually a user of the system. They typically follow a simple template: As a <user>, I can <action> so that <reason>

Quality Gatekeeper - Definition of Ready

Tells when an item is ready to be developed

- Is written down in the form: "*As a <user role> I want to <activity> so that <benefit>*"
- *INVEST* principles are met: Independent, Negotiable, Valuable, Estimable, Small and Testable
- Is mapped as a step in a business process diagram
- Business context and value are clear
- Is prioritized (for example based on MoSCoW)
- Conversations have taken place to clarify US so everyone (business, IT, Dev and QA teams) are aligned on what exactly to build
- Details are captured in acceptance criteria (functional and nonfunctional)
- Wireframes / mockups are drawn or reviewed for major US
- Is validated by business
- Test cases/scenarios are captured
- Meaningful and comprehensive test data is available
- Is estimated (at high level) and fits in a Sprint (recommended max size 8h to 12h, or 3 to 5 story points)

Quality Gatekeeper - Definition of Done

Tells when an item is 100% developed and tested. No more work is left to be done on this piece of functionality

- Code completed, adheres to IT guidelines and is published on DEV and QA environment
- Unit tests completed successfully by developers and confirmed by TL
- Test cases for acceptance criteria executed with success
- Reviewed and approved by EM
- Intermediate (preview) demo took place and a plan is in place to address its feedback during stabilization, before demo to business
- All planned feedback has been addressed and validated by the EM/QA
- All feedback issues associated with this US are closed
- Acceptance by the Customer

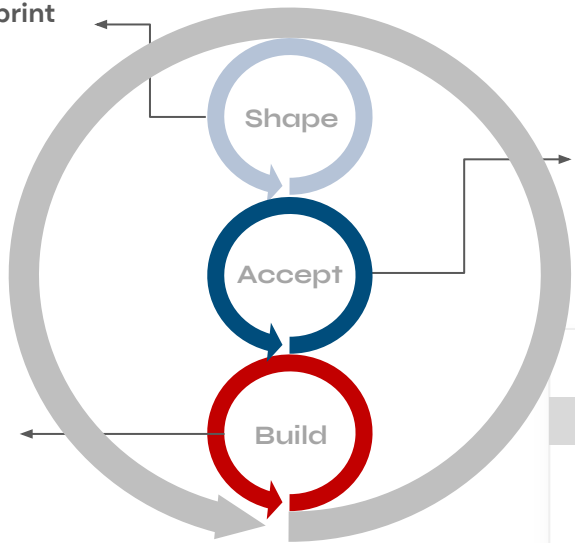
Template for writing effective user stories - Acceptance Criteria format

- Acceptance criteria: *"**Given that** <entry condition/context> **when I** <perform a specific action> **then** <expected reaction/response>"*
  - The User Story details are captured in acceptance criteria using **business user language**
  - Clearly **identify the impact** on the prior and later process steps (always referring to the base process flow)
  - Illustrate the **user experience scenarios** with pictures, screenshots or mockups
  - Provide **real sample data** (or as meaningful as possible): to be used during implementation, demo, and UAT
- Non-functional (aka quality) requirements
  - Impose constraints on the design or implementation (such as performance, availability, security or usability)

Sprint Delivery Model

Shape the next sprint

. New backlog
. Prioritize backlog
. Analyze and define the User Stories

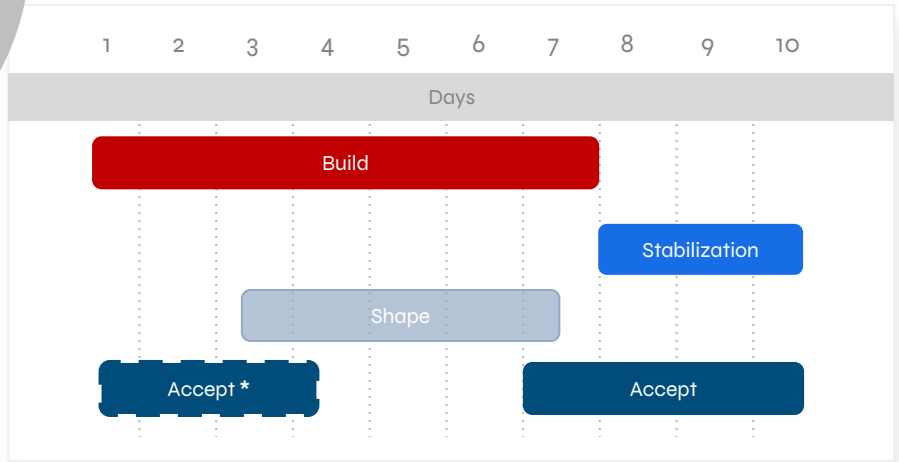


Acceptance of the previous sprint

Supported by OS, <customer> validates and accepts delivered User Stories

Deliver current sprint

User Story delivery



* The Accept track can be at the beginning of the iteration to accept stories from the prior iteration, or at the end of the iteration to accept stories just delivered, depending on your preferred delivery model.

outsystems

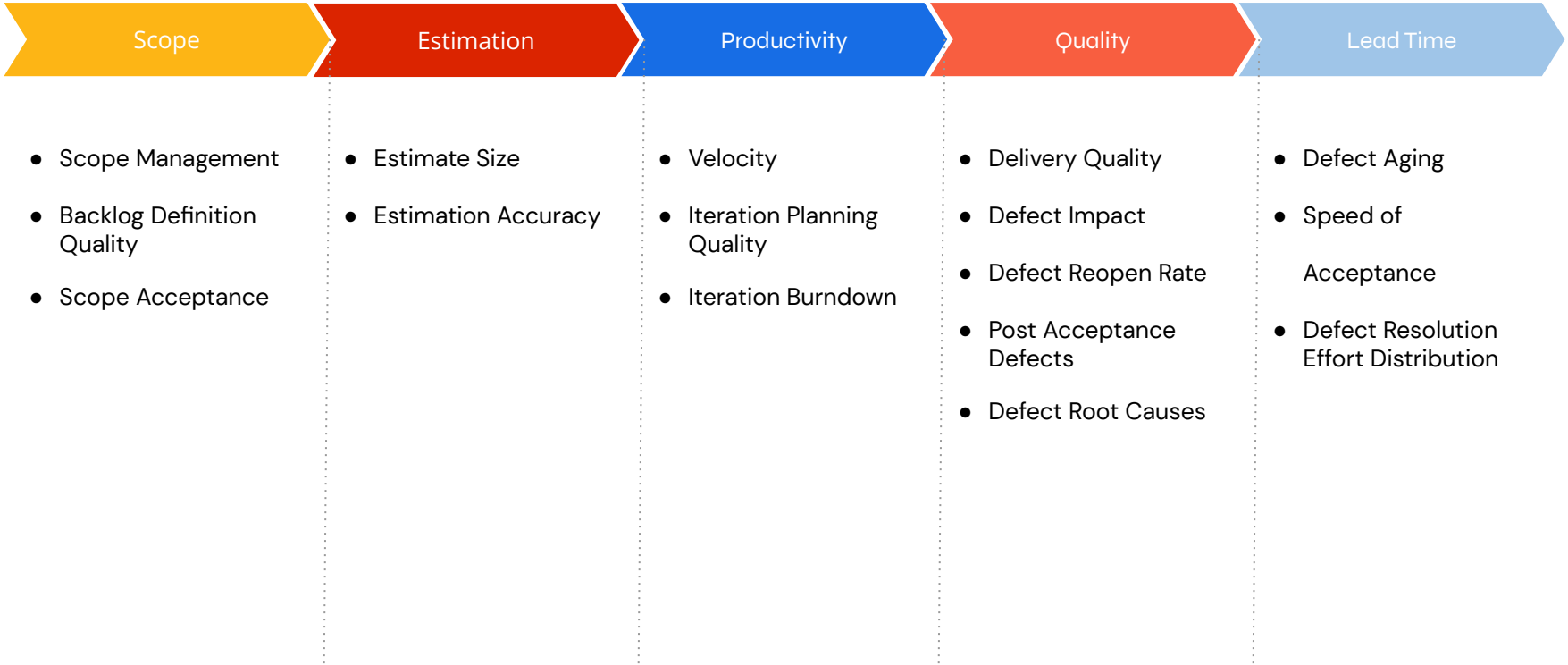
Metrics

Metrics - Goal

Identify and implement Project Management Metrics to help Project team having a clear and common approach on **project's status and health**, **foresee risks** and **assess team productivity** and **quality of work**.

- Identification of the Metrics aligned with the goals
- Build reports and dashboard based on the identified Metrics

Metrics - Overview



Note: This is the complete list of Metrics that can be tracked, each team needs to understand which ones are important and make sense for each particular case.

outsystems

Sprint Communication Plan

Sprint Ceremonies

Ceremony	Description	Attendees	Frequency
Daily Scrum	Review of completed work, plans, and impediments for the day	Mandatory: Full team Optional:	Daily
Backlog Refinement	Full team refinement of user stories to ensure that the team has stories DOR and ready for sprint planning.	Mandatory: Full team Optional:	2 - 3 / weeks
Sprint Planning	Full team plan and agreement on the user stories to be worked in the planned sprint.	Mandatory: Full team Optional:	At the start of each sprint

Team Ceremony

Team + Stakeholders

Sprint Ceremonies

Ceremony	Description	Attendees	Frequency
Sprint Retrospective	Discover what worked and didn't work during the sprint and the corrective actions to take for the next sprint.	Mandatory: Full team Optional:	At the end of each sprint
Sprint Demo	Demonstration of the user stories developed in the sprint and an opportunity for stakeholder review and feedback.	Mandatory: Full team + Key Stakeholders and Users Optional:	At the end of each sprint
Status Meeting	Discussion and review of completed work, upcoming activities, risks, issues and project status.	Mandatory: Full team + Key Stakeholders and Users	Weekly
Other	Defect triage, test planning, steering committee	For discussion	For discussion

Team Ceremony

Team + Stakeholders

Sprint Ceremony Schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
Week 01	Daily Scrum 9:00 - 9:15 Sprint Retrospective 9:30 - 10:00 Status Meeting 10:30 - 11:30	Daily Scrum 9:00 - 9:15 Backlog Refinement 10:00 - 11:30	Daily Scrum 9:00 - 9:15	Daily Scrum 9:00 - 9:15 Backlog Refinement 10:00 - 11:30	Daily Scrum 9:00 - 9:15
Week 02	Daily Scrum 9:00 - 9:15 Status Meeting 10:30 - 11:30	Daily Scrum 9:00 - 9:15 Backlog Refinement 10:00 - 11:30	Daily Scrum 9:00 - 9:15	Daily Scrum 9:00 - 9:15 Backlog Refinement 10:00 - 11:30	Daily Scrum 9:00 - 9:15 Business Demo 9:30 - 10:30 Sprint Planning 10:30 - 11:30

outsystems

Thank You!