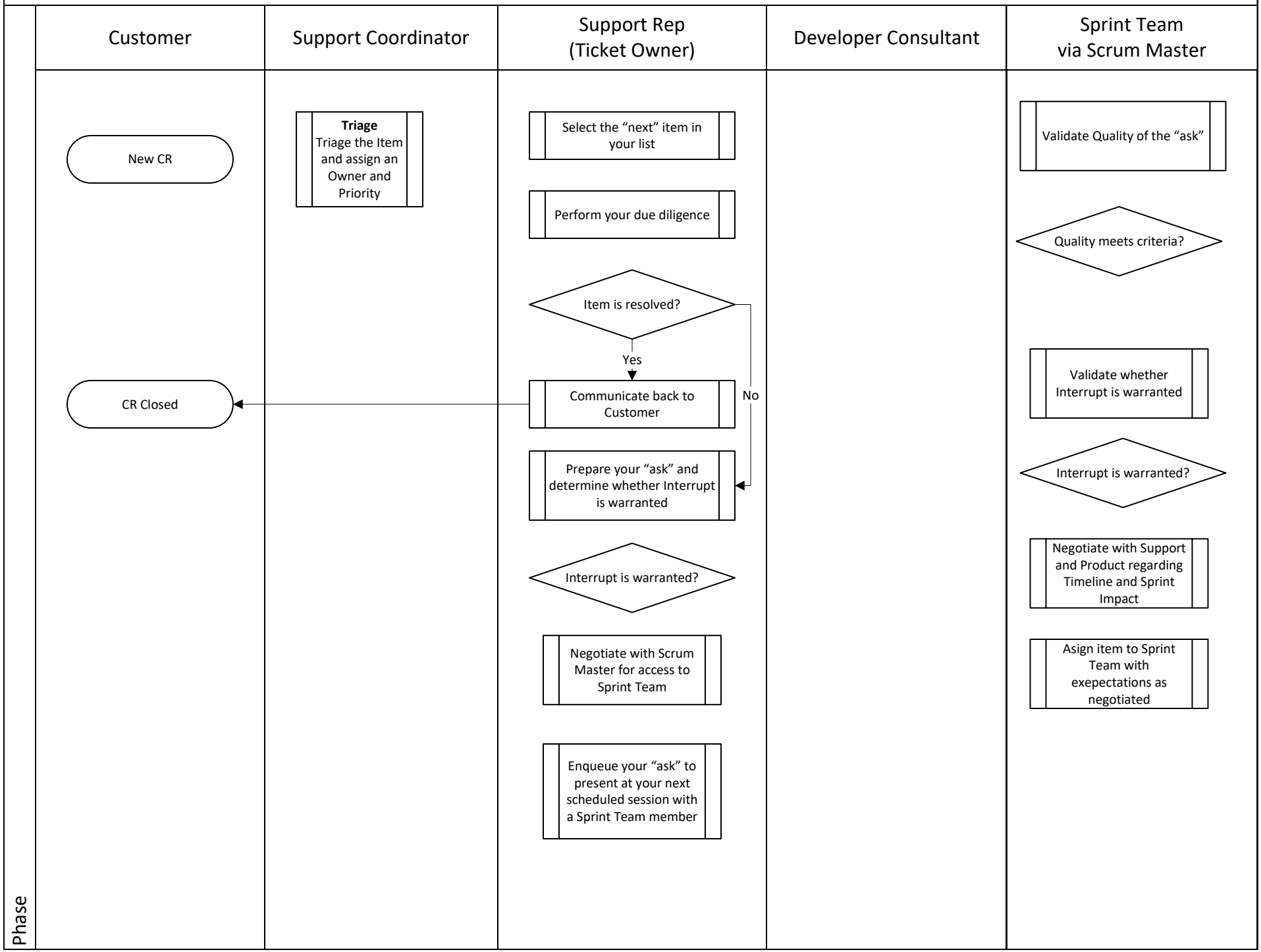
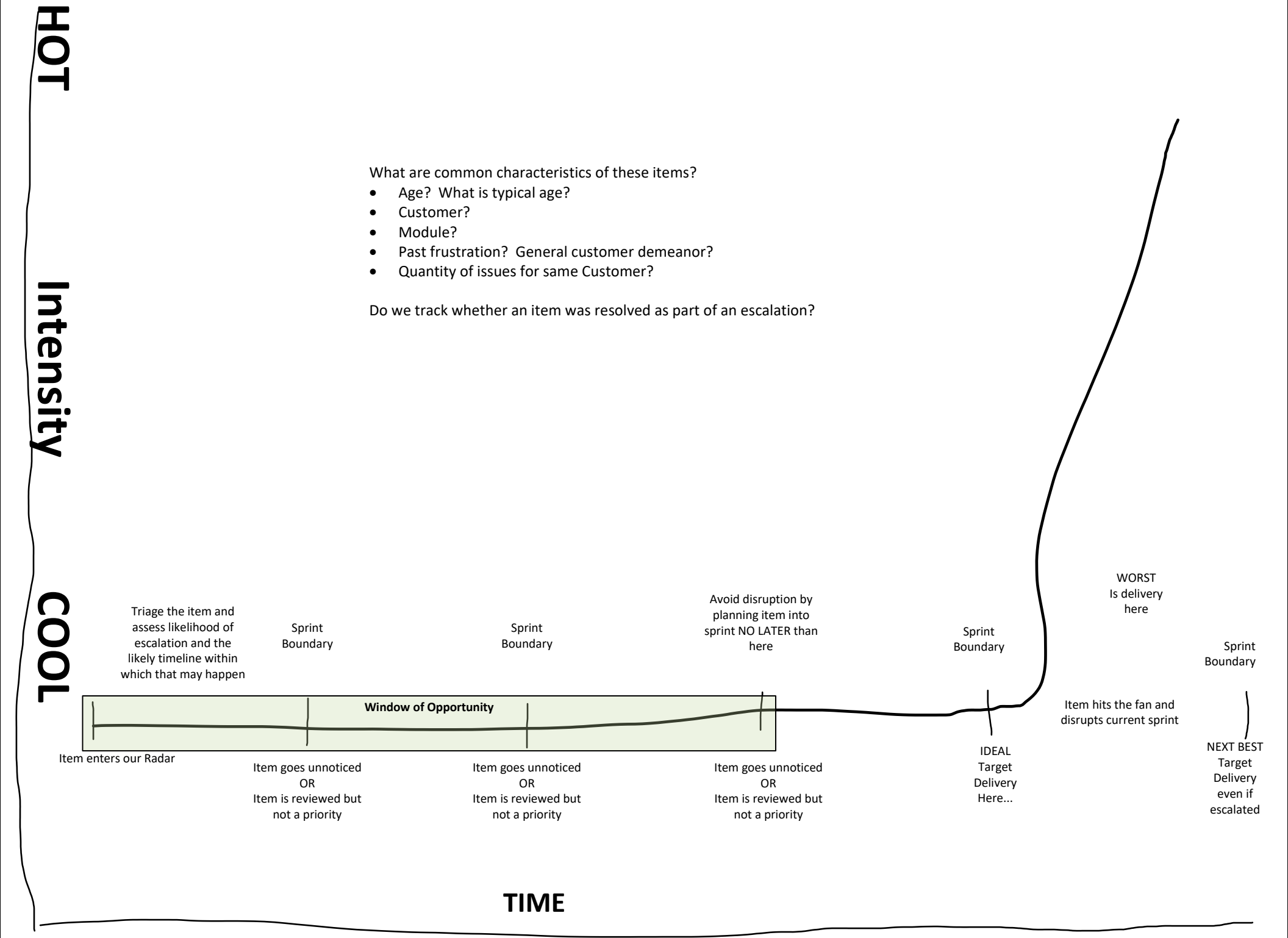
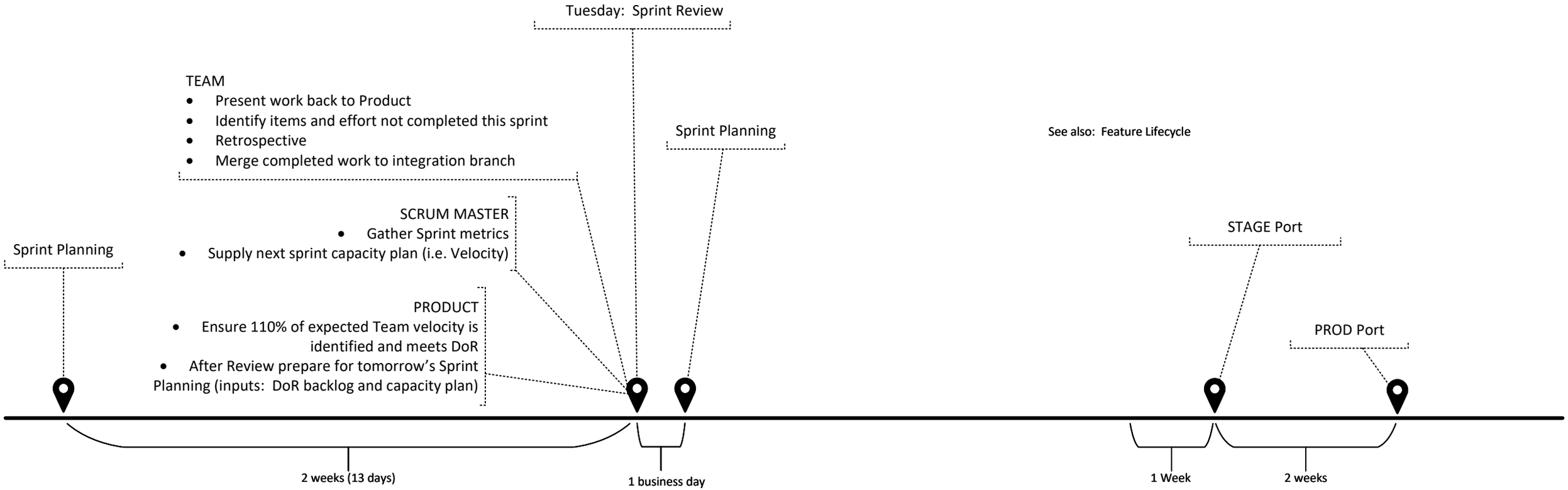
Support & CR Management



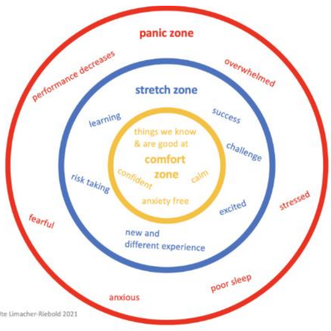

HOT
Intensity
COOL
What are common characteristics of these items?
Age? What is typical age?
Customer?
Module?
Past frustration? General customer demeanor?
Quantity of issues for same Customer?
Do we track whether an item was resolved as part of an escalation?
Triage the item and assess likelihood of escalation and the likely timeline within which that may happen
Sprint Boundary
Sprint Boundary
Avoid disruption by planning item into sprint NO LATER than here
Sprint Boundary
WORST Is delivery here
Sprint Boundary
Window of Opportunity
Item enters our Radar
Item goes unnoticed OR Item is reviewed but not a priority
Item goes unnoticed OR Item is reviewed but not a priority
Item goes unnoticed OR Item is reviewed but not a priority
IDEAL Target Delivery Here...
Item hits the fan and disrupts current sprint
NEXT BEST Target Delivery even if escalated
TIME

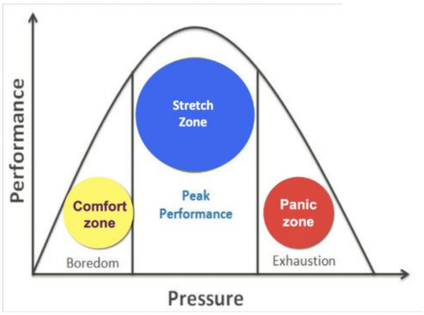


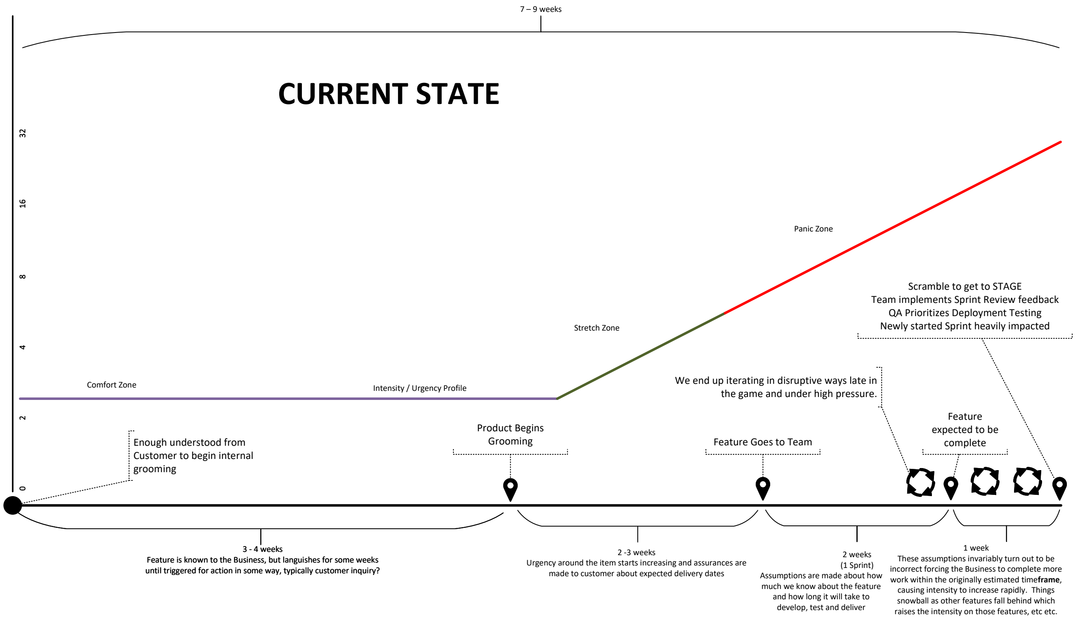
Product work prioritization:

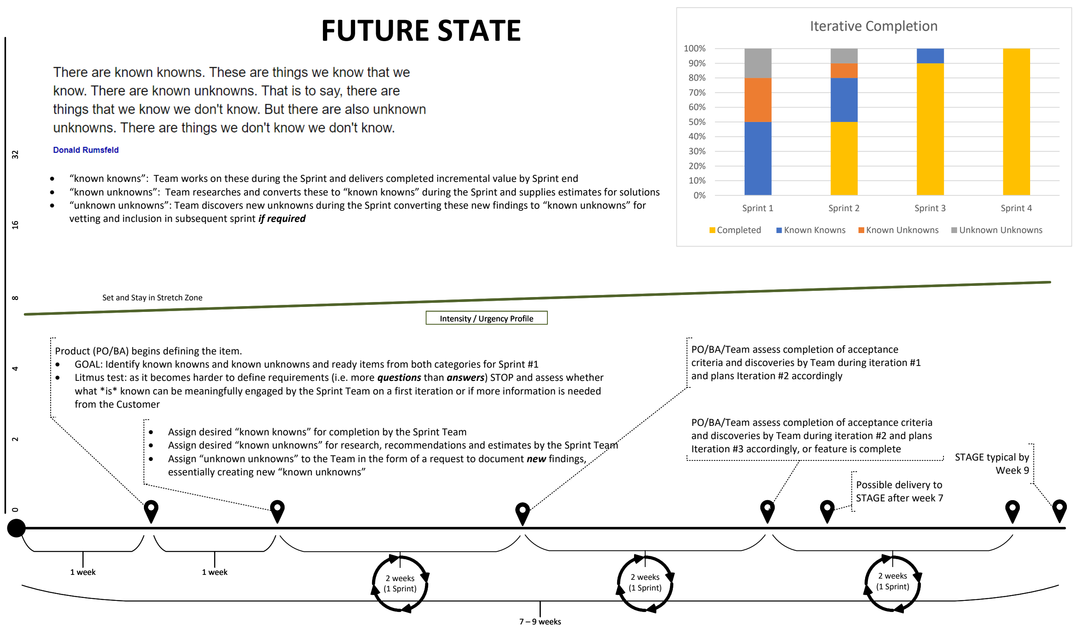
- Nothing is more important than being ready for the next Sprint (n + 1).
- Once the next Sprint is Ready, or on track to be ready, balance priorities among work related to sprint n+2 and other activities part of your role

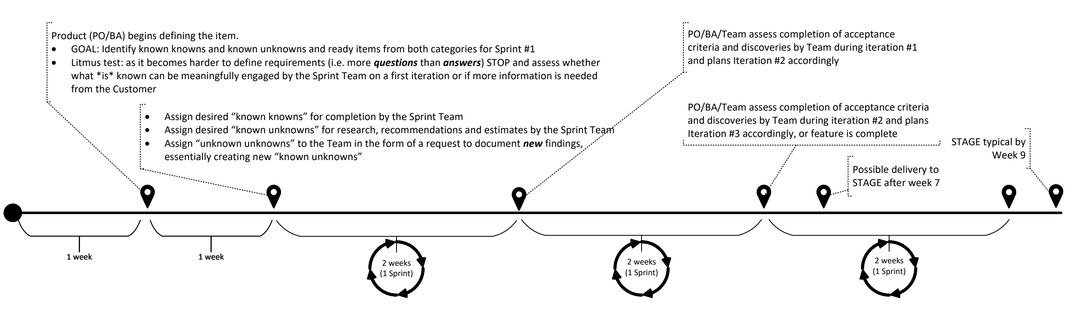
[See this article (and many others)]

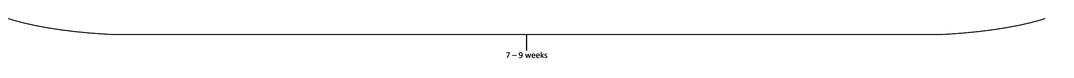












Do this if WiP Exceeded:

Unblock an item in the column.

If you need attention from another team member or colleague, ensure you get it.

If the item cannot be unblocked in this sprint, REMOVE it from the Sprint and ensure it is top of rank in the Product Backlog.

Do this if WiP Exceeded:

Finish an item already in the development column, and move it on to Code Review.

When selecting an item to finish, select from the items that have, at any time, already progressed the furthest right on the Board (example: Came from *Pending Bugs*)

Move an item BACK to TODO or into BLOCKED, respecting the WiP rules.

Do this if WiP Exceeded:

Complete the code review for an item in this column and move it to *Ready for QA* or Back to *In Development*

Do this if WiP Exceeded:

Team to discuss how to accelerate QA by utilizing more of the team's time (any role) to Test items *In QA*.

Do this if WiP Exceeded:

Team to discuss how to accelerate QA by utilizing more of the team's time (any role) to Test items *In QA*.

Complete QA and move to *Pending Bugs* or *Sprint Done*

Do this if WiP Exceeded:

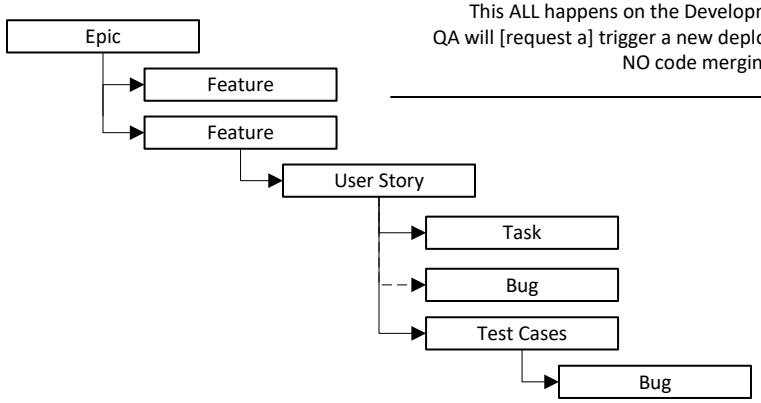
Move an item back to In Development, observe WiP rules.

Finish coding fixings for the "oldest" item already in this column.

For the assigned team member(s) DO NOT continue on with other development until these bugs are fixed and ready for testing.

Work in Progress (WiP) Limits	Not to Exceed Team Sprint Capacity	1 Sprint Item Per Developer	1 Sprint Item Per Developer	1 Sprint Item Per Developer	1 Sprint Item Per Developer	1 Sprint Items per QA Team Member	1 Sprint Item Per Developer	Unlimited
	TODO	BLOCKED	In Development	In Code Review	Ready for QA	In QA	Pending Bugs	Sprint Done Meets DoD
Feature 1 (Epic)								
Feature 2 (Epic)								
Feature 3 (Epic)								
Feature 4 (Epic)								
Maintenance ([Ongoing?] Epic)								

Focus on ***completing Features***. These are what ultimately get delivered to Customers.



Release Type

Sprint DONE to STAGE (Functional testing fully complete in DEV, ***create*** Release ***n+2***)

Sprint Review is Complete?

No

Special Circumstances?

No → Complete the Sprint

Yes

Yes

Stage UAT to STAGE (Urgent Customer issues fixed, re-port updated Release ***n+1***

STAGE to PROD (Port Release ***n+1*** to PROD for the first time)

STAGE_HF to PROD (Critical fixes addressed, re-port Release ***n*** to PROD)

Let Release ***n*** = Release currently on **PROD**

The ***Sprint DONE to STAGE*** process executes within the 6 business days between Sprint End and STAGE deployment date the following Thursday.

Track all time against ***SCTB-17 Release: Sprint to STAGE***

This same document will be re-used for STAGE to PROD
Set:
Product Name
Release Number
Merge Freeze Date and Time
Integration Code Freeze Date and Time
Port Tech Owner
Port Business Owner
Release artefacts folder (SharePoint)

Create the Next Release version number in Jira

Create an Instance of the Port Instructions from the Template
The **Port Tech Owner** owns oversight on remaining steps

Unlock Integration branch to accept Check-ins

Identify Release ***n*** branch (STAGE HF) and Merge ALL Changes from it through SOURCE to Integration

Identify Release ***n+1*** (STAGE) branch and Merge ALL Changes from it through SOURCE to Integration

Pull a TFS Report on all changes made within the branch structure that have occurred since the last Merge to Integration

Ideally this will always be a merge of the entire DEV branch since a branch is a feature, GOAL: no cherry picking.

Since the team is focused on completing sprint items for a feature and fully testing those items in the sprint, the feature branch can at least be expected to be stable.

Team should consider using shelf-sets as opposed to check-ins for any late sprint work that may destabilize the feature branch in DEV.

Review the Change Report with Contributors on branches not currently active with the Team. Select change sets / work items that must be included in the next STAGE release.

Merge all desired items and change sets from the specifc DEV branch(es) to the Integration Branch

An automation pipeline builds the Integration Branch. Devs resolve any build errors immediately.

Pull any Porting notes, scripts, resources together related to the work targeted to the Release and populate the ***Porting Instructions*** document

Execute the Porting instructions ***step-by-step*** on TEST, amend any discovered gaps in the instructions (Automation Deploys the Integration Branch to TEST).

Select Candidate items from the **DONE** Column in the Sprint and assign to Release Number to the Items (Jira)

No

Yes

Team (QA member) selects a build to deploy to TEST and runs the Release Test Plan (=Tests related to all items merged into REP-Integration)

Any Tests Fail?

Yes

Tester creates Bug from the Fail result (failing test info is automatically included in Bug)

Devs resolve any bugs directly on Integration

No

Additional items are "near" completion **AND** they ***must*** be in the next STAGE release?

Yes

Deliver completed Release Test Plan indicating all tests pass or documented releasable known issues.

Follow Branching Strategy to push Integration into Source and Branch Source to new branch Release ***n+2*** under Releases

Lock Integration branch until next release assembly

Execute Porting Instructions ***step-by-step*** on STAGE, amend any discovered gaps in the instructions

Merge Freeze Date is past?

No

As the Team gets better at ***finishing*** things ***before*** starting new things, the list of items that are "close" will become very short. Items will ***either*** be ***done*** or not meaningfully started. The Team can help themselves by ensuring this is the case as the Sprint progress – see ***Sprint Board*** slide.

Team (QA member) selects a build to deploy to TEST and runs the Release Test Plan on STAGE

Any Tests Fail?

Yes

Tester creates Bug from the Fail result (failing test info is automatically included in Bug)

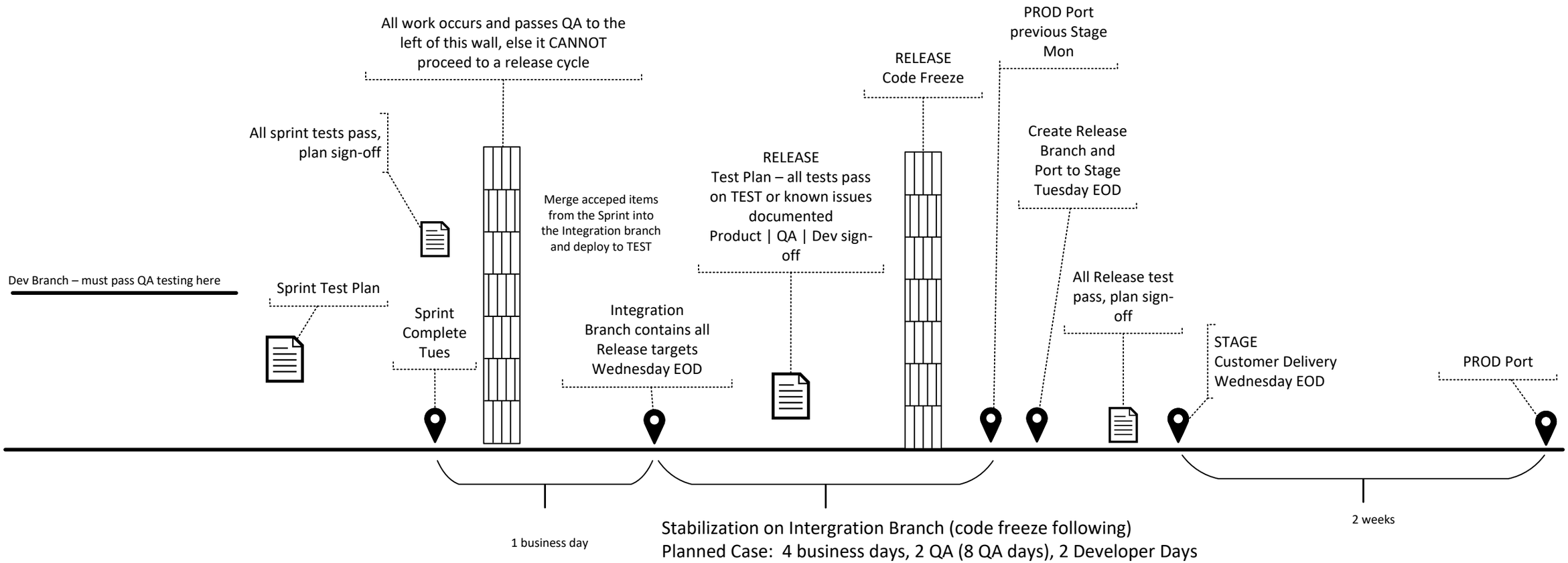
Devs resolve any bugs directly on Release ***n+2*** Branch

No

Deliver completed Release Test Plan indicating all tests pass or documented releasable known issues.

DONE RELEASE

Complete development and testing on candidate items in the DEV branch in which they exist


All work occurs and passes QA to the left of this wall, else it CANNOT proceed to a release cycle
PROD Port previous Stage Mon
RELEASE Code Freeze
All sprint tests pass, plan sign-off
Create Release Branch and Port to Stage Tuesday EOD
Merge acceped items from the Sprint into the Integration branch and deploy to TEST
RELEASE Test Plan – all tests pass on TEST or known issues documented Product | QA | Dev sign-off
Dev Branch – must pass QA testing here
Sprint Test Plan
All Release test pass, plan sign-off
Sprint Complete Tues
Integration Branch contains all Release targets Wednesday EOD
STAGE Customer Delivery Wednesday EOD
PROD Port
1 business day
2 weeks
Stabilization on Intergration Branch (code freeze following)
Planned Case: 4 business days, 2 QA (8 QA days), 2 Developer Days
Best Case: 1 business day, 2 QA (2 QA Days), .5 Developer Days
Expected Case: 2 business days, 2 QA (4 QA Days), 1 Developer Day
Worst Case: 4 business days, 2 QA (8QA days), 2 Developer Days