ImageX Scrumfall Methodology

MODIFICATION DATE	Mon Oct 28 2013
CREATOR	Adrian Jones, Senior Business & UX Analyst Lead, ImageX
SOURCE FILE NAME	ImageX_Scrumfall_v1.graffle
DESCRIPTION	This is a collection of diagrams for a hybrid Waterfall-Agile, or Scrumfall, methodology for implementation in a services agency business model.

IMAGEX

Scrumfall Overview

SCRUMfall

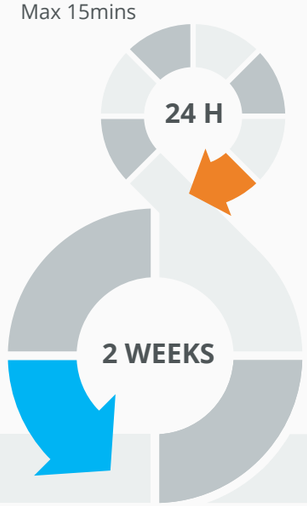
Water...

Requirements Analysis
Project Planning
UX Design
Information Architecture
Wireframing

SCRUM

Iterative development

Daily Standup
Max 15mins



Product Backlog
Prioritized features desired by client

Sprint Backlog
User stories committed to for sprint

Release

PRODUCT BACKLOG

SPRINT BACKLOG

SPRINT

POTENTIALLY SHIPPABLE PRODUCT

...Fall

Controlled production release

Agile Board Definitions

Issue Types

The following are the types of issues that are typically documented and tracked on your Agile board, which could be analog (e.g. whiteboard, sticky notes), or online (e.g. JIRA, Pivotal Tracker).

Epic:
An epic is a feature or requirement that is either too big to be completed in a single sprint, or that has too many unknowns to have its level of effort accurately estimated. Another good rule-of-thumb is that if a User Story will require more than 2 days to complete it should probably be considered an epic and broken down into smaller stories.

Epics can be broken down any time, but Sprint Planning and Backlog Grooming are good opportunities to do so.

Story:
A story, or user story, is a description of a requirement with particular attention to the end user's point of view. It describes something a user is hoping to do with, or get from, the system.

Many teams have learned the benefits of writing user stories in the form of: "As a <type of user> I <want/can/am able to/need to/etc.> so that <some reason>." But it is not necessary that a user story be written this way.

A story is granular enough that it can be completed in a single sprint, and ideally in less than 2 days of effort.

Task:
A task is something a development team member needs to do on the way to completing a Story. After a dev commits to a story for the upcoming sprint they can begin to break down that story into tasks. Additional tasks may also be created while the story is being worked on during the sprint, as required.

Tasks are given effort estimates in hours.

Bug:
A bug, or defect, is a problem with the system that needs to be fixed. A bug can be created by anyone, although they are typically logged by QA. A bug is often created if a defect is found that relates to an already completed piece of functionality (for example, if new work introduces a bug into previous work). If a defect is found on a story while the sprint is still on, it should be noted in the story and assigned back to the dev team member. For example, if a bug is found while a story is in QA, the defect should be noted in the story's description and then assigned back to the dev. In this case a separate bug issue is not created.

Sometimes, if a small bug is found for a story during its demo to the Product Owner, we might choose to accept the story itself and log a separate bug to tackle the minor defect. If the bug is medium to large, though, the story should be rejected.

Issue Metadata

The following are some of the details and metadata that should be attached to issues on the Agile board.

Definition of Done (DoD):
Before a story is started, it should have the following:
1. A clear title and description.
2. Effort estimate (relative points are best, but hours can also be used).
3. Test cases.
4. Acceptance criteria.

If a dev completes all of a story's tasks, the story passes all the test cases, and finally, it meets all the acceptance criteria, it is said to meet the DoD and is production ready.

Components:
Sometimes there are Stories that span across multiple epics which all relate to a common feature. For example, a user registration system may be complex enough that its user stories belong to more than one epic. In this case we can relate all of these stories together by associating them with a Component, (also called a Theme). Examples of possible components are "User Registration", "Database", "User Interface".

Both epics and stories can be associated with a Component.

Versions:
Stories can be assigned to versions, e.g. 1.0, 2.4. This isn't needed on every project, but can be useful on projects where the development is broken up into multiple phases because of budget constraints, scheduled releases, etc.

Scrumfall Breakdown

IMPORTANT: ALL ACTIVITIES AND DELIVERABLES ARE *AS NEEDED*, MEANING DON'T DO THEM IF THE PROJECT DOESN'T NEED THEM!

LIFE CYCLE PHASES

	WATERFALL ACTIVITIES	AGILE (SCRUM) ACTIVITIES	ROLES INVOLVED	DELIVERABLES
INFORMATION DISCOVERY	Project Planning; Business Requirements; Schedule; Risk Analysis; Scope Definition	Brainstorming; High-level Requirements; Elevator Pitch; Product Box; Innovation Games	Business Analyst; Project Manager; UX Designer	Project Plan; Business Requirements Document
REQUIREMENTS ANALYSIS & IA	Functional Requirements; Content Types; Taxonomy; Sitemap; User Roles & Permissions; Non-functional Requirements; Technical Design	Low Fidelity Wireframes; Paper Prototypes; Affinity Diagramming; Use Cases; Process Flow Diagrams; UML Models	Business Analyst; Project Manager; UX Designer; Dev Lead	System Requirements Document; Functional Requirements; Non-functional Requirements; User Requirements; Products of the Agile Activies, e.g. wireframes; Technical Design Document
DESIGN	Define Style & Branding Guidelines; Design Visual Mockups	Higher Fidelity Wireframes; Responsive Layouts; Interactive Designs	UX Designer; Graphic Designer; Project Manager	Products of both the Waterfall and Agile Activities
DEVELOPMENT	Build the Initial Technical Architecture & Infastructure	Break Down Epics into User Stories; Work on Completing User Stories; Use Features Module; Continuous Integration	Client Product Owner; Business Analyst/ Product Owner; Project Manager/ Scrum Master; Dev Lead; Developer(s); Content Creator(s); Quality Assurance	Potentially Shippable Product Increments
CONTENT CREATION	Create and/or Migrate Site Content			
QUALITY ASSURANCE	Match Completed Features with Functional Requirements; Regression Tests	User Stories Go Through QA to Pass the Definition of Done; Product Owner Accepts/Rejects Stories		
DEPLOYMENT/ LAUNCH	Checklist for Launch		Client; Project Manager; Dev Lead	Completed Site Launch
MAINTENANCE	Maintenance Tasks		Developer(s); Quality Assurance	Maintenance Releases

WATER...

SCRUM

...FALL

Feature Life Cycle

LIFE CYCLE PHASES

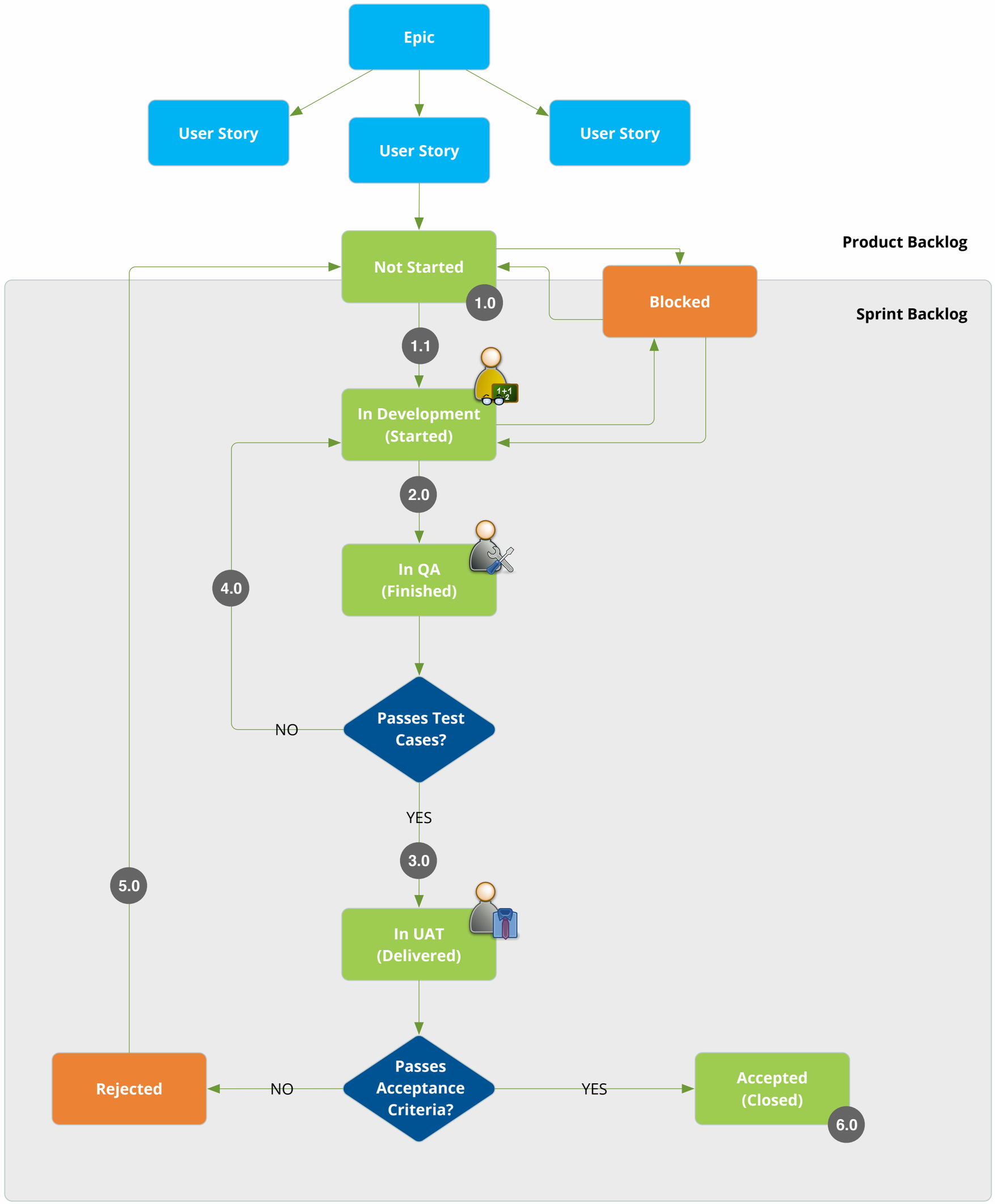
	ROLES INVOLVED	DESCRIPTION	FEATURE/FUNCTIONALITY
INFORMATION DISCOVERY	Business Analyst	Business Analyst identifies a business requirement.	Feature Inception
REQUIREMENTS ANALYSIS & IA	Business Analyst UX Designer	Business Analyst turns the business requirement into one or more functional and/or user requirements. UX Designer creates wireframes, prototypes, interaction diagrams, etc. as needed to further define and understand the feature.	Feature Definition Feature IA/UX Assets
DESIGN	UX Designer Graphic Designer	UX Designer supports Graphic Designer as visual mockups are created to illustrate what the feature will look like when styled.	Feature Visuals
DEVELOPMENT	Business Analyst/ Product Owner Dev Lead Developer(s)	Business Analyst turns the feature into an Epic or User Story and inserts it in priority order into the Product Backlog. Epics are broken down into User Stories. User Stories are given point estimates, test cases, and acceptance criteria. Developer breaks down User Story into Tasks and works on completing them.	Epic User Story User Story User Story has Task Task Task broken down into Points Test Cases Acceptance Criteria
CONTENT CREATION	Content Creator(s)	If the feature involves content creation, this content is added.	
QUALITY ASSURANCE	Quality Assurance Client Product Owner	Feature goes through QA test cases. If all test cases pass, the feature can be demoed. Feature is evaluated by Client Product Owner against acceptance criteria. If all acceptance criteria pass, feature is accepted as done.	
DEPLOYMENT/ LAUNCH	Dev Lead	Code for accepted User Stories is pushed from staging to production server.	Completed Feature
MAINTENANCE	Dev Lead Developer(s)	Feature is maintained and improved after being pushed to production.	Improved Feature

WATER...

SCRUM

...FALL

User Story Workflow



Assignments:

1. Dev team member commits to a user story during Sprint Planning.

1.1 Story marked as "In Dev/started" when dev member begins work on it.

2. Story completed by dev and marked as "In QA/finished". Assigned to QA.
3. Story passes QA and marked as "In UAT/delivered". Assigned to Product Owner.
4. Story reverted back to "In Dev/ started". Assigned back to dev team member.
5. Story reverted back to "Not Started". Assigned back to dev team member. Likely immediately added to the next sprint.
6. Story meets our Definition of Done. Assigned to whoever created the story (likely a BA) and closed.

Issue Types:
1. Epic
2. Story
3. Task
4. Bug

*See "Agile Board Definitions" on pg.3