AGILE – SCRUM

* Agile tốn nhiều công sức hơn để xây dựng và phát triển
* Khắc phục điểm yếu của Waterfall (tốn nguồn lực để chỉnh sửa/phát triển khi phần mềm đã phát triển mạnh)
* Agile giúp chúng ta bẻ nhỏ công việc, dễ theo sát và thay đổi chỉnh sửa, thay đổi plan cũng như dễ tính toán hơn

AGILE TEAM

* Khó xác định vai trò vì mọi người phải làm tất cả những gì có thể để hiện thực công việc
* Mọi công việc phải thực hiện chung, xuyên suốt :analysis, design, implementation và testing
* Chất lượng chính là trách nhiệm của cả team -> có bug là cả team chịu trách nhiệm xử lý
* Cả team phải nên làm việc gần nhau, phản hồi và xử lý nhanh

SCRUM

* Mọi developments đều được viết doc, hiện thực và test mỗi chu kì sprint -> có thể trình chiếu được ngay
* Mỗi sprint từ 2-4 tuần
* Thời gian này nên cố định -> tạo nhịp độ thực hiện

SCRUM BOARD

* Có 1 board để cả team lên ý tưởng, sắp xếp, thảo luận task trực tiếp và tức thì
* JIRA có tính năng này

BACKLOG

* Backlog là requirements -> nên có meeting để xác định backlog nếu chưa có
* Sau mỗi sprint thì sắp xếp lại
* xác định trong plan meeting
* Product Owner quản lý backlog

USE CASE / USER STORIES

* Giúp cả team xác định được đâu là mục tiêu, kỳ vọng của dự án bằng cách sử dụng 1 câu chuyện nói lên mong muốn của user

DONE

* Done hay không phụ thuộc vào mục tiêu ban đầu : dự án thành phẩm hay sản phẩm mẫu ?
* Xác định loại Done ngay từ đầu và giống nhau cho tất cả thành phần của project -> chỉ hoàn thành SPRINT khi DONE.

EPICS

* Epic là bức tranh lớn hơn, cần nhiều sprint để thực hiện
* Epic sẽ xảy ra các vấn đề phụ (sub-issue) không đc phát hiện khi phát triển sprint riêng lẻ -> thêm task vào sprint
* Giá trị của sản phẩm chỉ đạt được sau khi hoàn thành toàn bộ sprint cấu thành epic

Participants in a SCRUM process

* Product owner: tập trung vào tính năng, lịch trình, ngân sách cũng như ưu tiên lại thứ tự backlog, thêm tính năng vào mỗi sprint
* SCRUM master: tập trung quản lý nhóm, loại bỏ các vấn đề, gắn kết và tăng năng suất của cả team
* Các thành viên còn lại: tăng khả năng tự sắp xếp, phân bổ đầy đủ ở mọi tính năng. Chỉ thay đổi nhân sự khi chuyển tiếp giữa 2 sprint

SCRUM

* Chuẩn bị Product Planning -> đưa vào Backlog và lên Sprint Planning cho phase Sprint 1
* Khi kết thúc phase Sprint 1 -> lên Sprint Planning cho phase kế (Được lấy từ Backlog)
* Khởi đầu Sprint 2 với việc Review Sprint 1 và Sprint 1 Retrospective
* Lặp lại cho đến khi hết Backlog

TEAM SIZES

* Một team SCRUM nên có 5-9 người (có thể hơn 1 chút), nhiều hơn thì tách ra thành các team nhỏ hơn và không nên có quá ít thành viên
* Các cuộc họp phải ngắn và xúc tích -> tăng hiệu quả

PLANNING AND ESTIMATION

* Tính toán thời gian cho các công đoạn -> ra thời lượng dự kiến cho toàn project

SPEED = VELOCITY

* Tốc độ của 1 team phụ thuộc vào thời gian thực hiện 1 sprint -> từ đó xác định được hiệu suất của team ( do các sprint nên có gần như cùng thời gian hoàn thành -> có thể theo dõi hiệu suất qua từng sprint)
* Không tính toán dựa trên 1 cá nhân mà phải là tập thể

VELOCITY

* Chỉ tính các phần công việc ĐÃ HOÀN THÀNH
* Nên chia nhỏ các công việc để phù hợp với yêu cầu trên

VELOCITY ‘WILL’ FLUCTUATE

* Velocity sẽ thay đổi qua các sprint -> nên lấy trung bình để nắm tốc độ của team

PRODUCT PLANNING

* Gần như mọi sản phẩm đều cần đầu ra tổng quát trước khi triển khai
* Chủ dự án sẽ có buổi họp xác định overall size và goal của dự án
* Agile phù hợp với các dự án luôn có cập nhật, thay đổi trong tương lai
* Xác định backlog lúc khởi động dự án, sau đó là risk (dựa vào số liệu trong quá khứ, số nhân lực)
* Sử dụng buffer để quản lý dự án
* Phải có estimates time cho toàn bộ issues trong backlog nhanh nhất có thể thông qua product planning
* Mỗi sprint planning đều phải lướt qua backlog và xác định lại estimated time

SPRINT PLANNING MEETING

* Công việc rút ra khỏi backlog khi sprint cho đến khi lắp đầy estimated time/story points
* Lập thành sơ đồ phát triển ứng dụng
* Sử dụng story points hoặc days để ước lượng thời gian hoàn thành toàn bộ items để kiểm tra xem có phù hợp mục tiêu ban đầu hay không
* Cả team cùng thực hiện công việc này

INCREMENT

* Là tổng công việc đã hoàn thành trong sprint, mọi công việc phải đạt được status DONE (có thể đưa ra sử dụng ngay sau khi DONE)

STORY POINT

* Sử dụng thay cho days, đánh giá dựa trên độ phức tạp của công việc
* Qua vài task sẽ tính được thời lượng trung bình dựa trên story points
* Thực hiện hiệu quả nhất do không có thời gian trống như estimated time in days

DAYS

* Sử dụng ‘ideal days’ để tính toán thay cho story points
* Ideal days là ngày không có thời gian bận

PLANNING POKER

* Mọi người cùng tham gia ước lượng story points cho mỗi story
* Nếu có chênh lệch lớn -> thảo luận lại

SPRINT PLANNING MEETING

* Dành nhiều thời gian cho việc planning (>20% time)
* Cần nỗ lực lớn từ Product Manager và SCRUM Master để theo dõi tiến độ
* Lúc này Product Owner sẽ quản lý các item trong backlog

PLANNING PRIORITY

* Đảm bảo các user story mang lại nhiều giá trị nhất phải được làm trước và Done (đem lại lòng tin cho khách hàng)
* Các tính năng basic phải làm trước các tính năng ‘nice-to-have’

DAILY SCRUM MEETING

* Mọi thành viên phải tham gia, không dài quá 15p, mọi người đứng trao đổi nhanh về tổng quan dự án (không bàn sâu chi tiết)
* Có 3 vấn đề mỗi người cần trình bày : Hôm qua làm gì, hôm nay sẽ làm gì và có vướng mắt gì không?
* Nếu có vấn đề thì tách ra xử lý riêng, ko được bàn mọi issue không liên quan đến hôm qua và hôm nay
* Dành cho team member để nắm tình hình
* Không record, ko lên plan, ko bàn giải pháp trong buổi này

RETROSPECTIVE MEETING

* Sau mỗi sprint được hoàn thành thì tổ chức 1 retrospective meeting
* Buổi này có toàn bộ thành viên từ Product Owner, Scrum master đến team member
* Không dài hơn 60p
* Review lại những điểm tốt và chưa tốt, không chỉ trích cá nhân
* Sử dụng Sprint metrics như : velocity over time, % item chưa hoàn thành/bị xóa,% scope change, amount of reopen issues

PRODUCT REVIEW MEETING

* Mời mọi stakeholder, ko dùng slide, kể lại toàn bộ những gì đã làm trong sprint vừa rồi
* Cần thiết để nhận phản hồi cho sprint kế tiếp