Module 8

AGILE Concepts

Course Agenda & Organization

- Module 1: Introduction, Definitions, Concepts
- Module 2: Product Scope: Concepts
- Module 3: Project Scope: Deliverables & WBS
- Module 4: Schedule Management
- Module 5: Risk Management
- Module 6: Resources, Costs,
- Module 7: Communication, Note de cadrage, Program and Portfolio management.
- Module 8: AGILE concepts

- **Fil Rouge**: Tender Yakasserole (YAKA*)

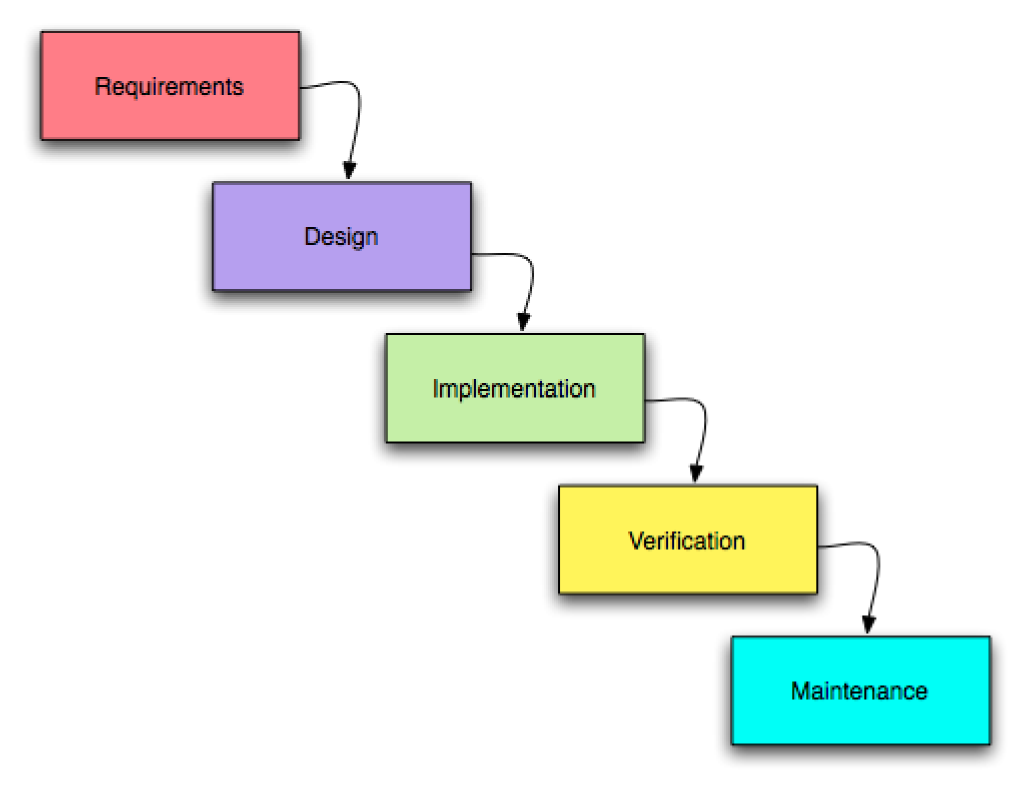
- Evaluations:
  - QCM: Partiel le 1er Avril
  - YAKA*: GO/NOGO
  - Retour sur PROPAL YAKA*

SOFTWARE Development : The Software Crisis

- WATERFALL comes from Manufacturing and Building
- SHOWS a working product at 2/3 of the way
- 1994 Report on Software Development Projects
  - 16 % Successfully finished ontime within budget and scope
  - 85 % End up as Failures…

For SOFTWARE Developement : CHANGES are UNAVOIDABLE

- Technical changes
- Competitive landscape
- Requirements changes



AGILE To the Rescue :

a new approach based on EMPIRISM and SMALL INCREMENTS

AGILE MANIFESTO

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have to come to value:

Individuals and interactions over processes & tools

Working software over comprehensive documentation

Customer collaboration over contract negociation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more

AGILE PRINCIPLES (x12)

1. Satisfy the customer through early and continuous delivery of valuable Software
2. Welcome changing requirements, even late in development
3. Deliver Working Software Frequently (Weekly or Monthly) With Focus on Shorter Timescale
4. Business people and developers work together daily throughout the project
5. Build projects around motivated individuals
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

AGILE PRINCIPLES (x12) …

7. Working software is the primary measure of Success
8. AGILE process promotes sustainable Development
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity : the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams
12. At regular intervals, the team reflects on how to become more effective, and then tunes and adjusts its behavior accordingly.

AGILE “Methods”

SCRUM

KANBAN

XP

SCRUM Framework

– Lightweight FRAMEWORK often confused with AGILE itself

– Presented in 1995 by Ken Schwarber and Jeff Sutherland

– Simple to understand, Difficult to Master : Scrum Guide available on Scrumguides.org: 20 pages

– Not prescriptive

– Based on EMPIRICISM opposite of PREDICTIVE

Widely used: Scrum is a simple set of principles and practices that helps teams deliver products in short cycles, enabling fast feedback, continual improvement, and rapid adaptation to change.

SCRUM defines:

- 3 Roles: Product Owner, Scrum Master, Development Team
- 3 Artifacts (or Deliverables): Product Backlog, Sprint Backlog, Increment
- 5 Events: Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective

SCRUM Values

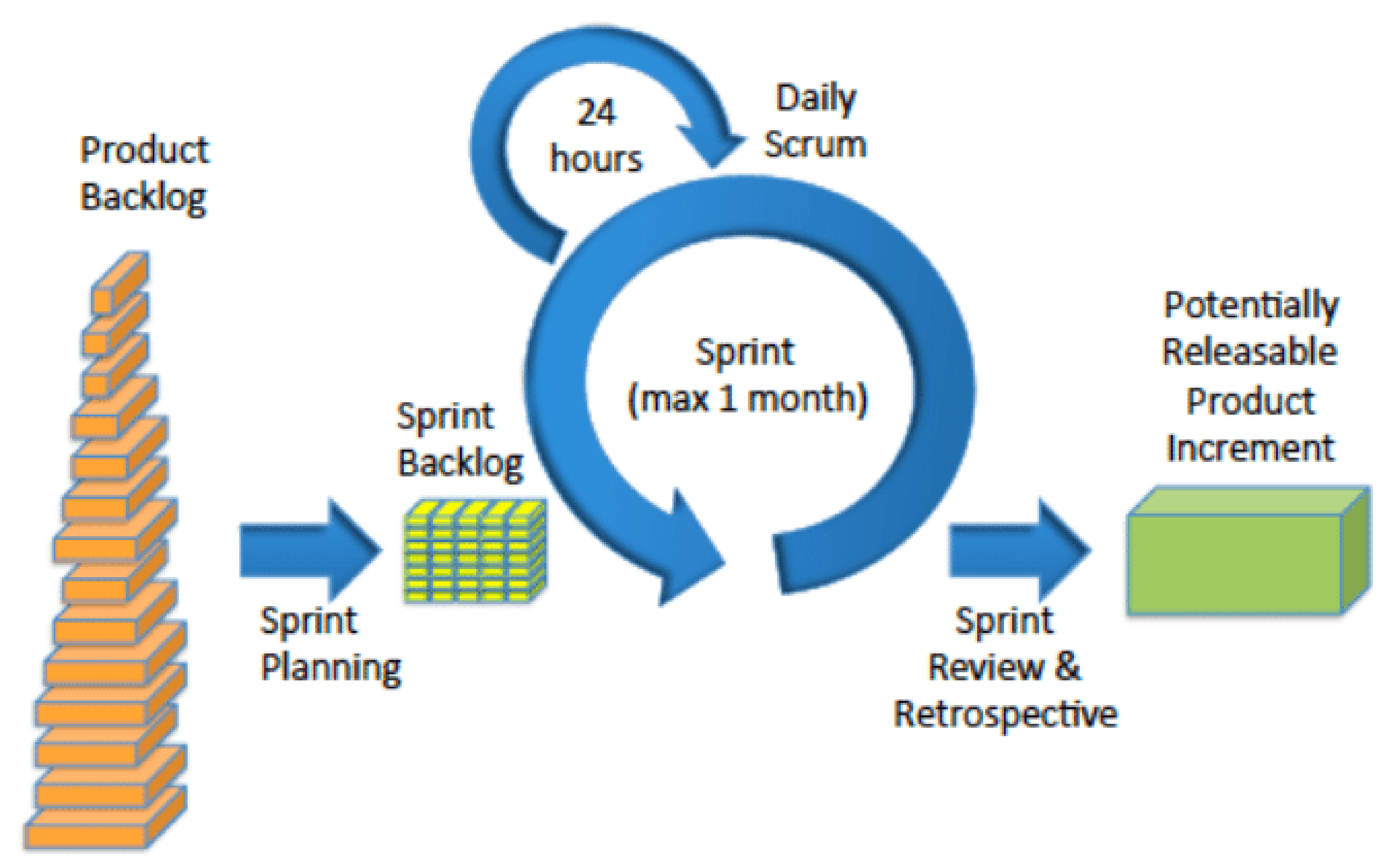
Three PILARS: SCRUM based on TIA

- Transparency: at every stage through Information radiators : Sprint status, Bill status, defect count. No need for Status review, everything is displayed.
- Inspection: you can inspect only what is transparent
- Adapt: you can adapt only what you can inspect

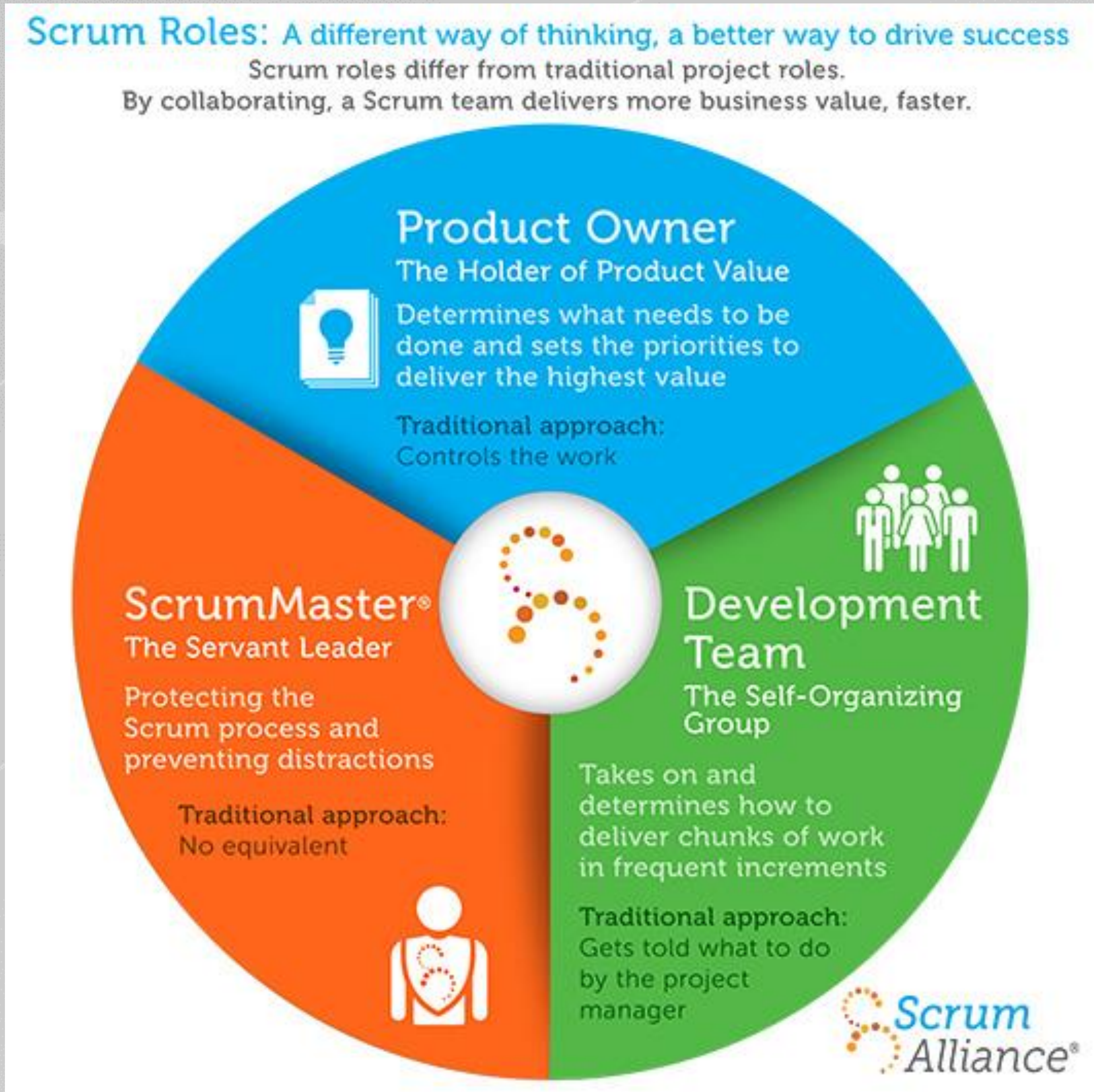
Five GOAL VALUES

- Focus (Timebox)
- Respect
- Openness
- Courage (To adapt)
- Commitment (To the AGILE Principles)

SCRUM Process OverView



SCRUM ROLES



Scrum Roles: A different way of thinking, a better way to drive success

Scrum roles differ from traditional project roles.
By collaborating, a Scrum team delivers more business value, faster.

Product Owner
The Holder of Product Value

Determines what needs to be done and sets the priorities to deliver the highest value

Traditional approach:
Controls the work

ScrumMaster®
The Servant Leader

Protecting the Scrum process and preventing distractions

Traditional approach:
No equivalent

Development Team
The Self-Organizing Group

Takes on and determines how to deliver chunks of work in frequent increments

Traditional approach:
Gets told what to do by the project manager

SCRUM Artifacts

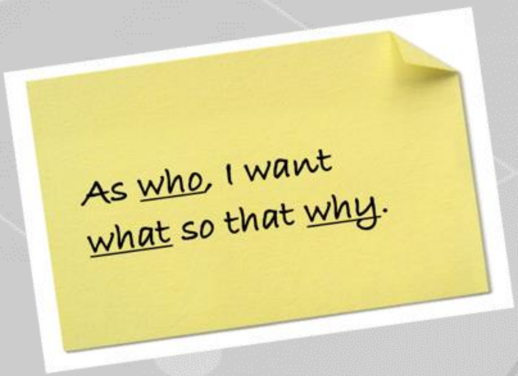
- Product Backlog is the TO-DO LIST
  - Contains Requirements, Enhancements, Defects, User stories, New features ideas
  - ICEBERG: item closer to the top are smaller (easier to estimate and implement)
  - Larger items are divided into smaller : GROOMING, or refinement
  - Items are organized by priority: higher priority has more fine grain
  - Product backlog items are estimated
- Sprint Backlog:
  - Subset of product backlog items selected for the Sprint
  - The development team plan
- Increment:
  - The product at the end of each Sprint
  - Current « done » version of the Software
  - A vertically sliced portion of the system (end to end functionnality)

User Story

A user story is a simple and concise way of expressing product requirements.

It is the DeFacto standard for expressing requirements in AGILE (NOT mentionned specifically in SCRUM)

3 Cs : Card / Conversation / Confirmation



As who, I want what so that why.

Example:

User Story:

« As a potential Fitness club member, i should be able to sign-up as a trial member and print temporary badge so i can try fitness center facilities »

Details / How to verify it is done:

1) I should be able to enter my name, phone number, and email addres so i can sign up as a trial member
2) The webpage should generate a membership badge with scannable barcode that i could print
3) The webpage should email a temporary badge ID with scannable bar code
4) The membership should be valid for 5 calendar days from the date of signup
5) The temporary badge should display « Trial Member »

User Story

Example 2: YAKASSEROLE

User Story:

« En temps que client premium, je dois pouvoir réserver en ligne et payer un atelier de cuisine pour 4 personnes pour gagner du temps par rapport à une réservation téléphonique»

Details / How to verify it is done:

1) Pour une sélection de dates, Je dois pouvoir visualiser les types d'atelier, leur localisation géographique et la disponibilité
2) La page WEB me donne accès à un formulaire de réservation
3) Le formulaire me permet de donner les emails des 3 autres participants pour qu'ils soient informés de ma réservation
4) Ce formulaire de réservation permet de confirmer ma réservation et aussi effectuer le paiement en ligne
5) Je reçois par mail la confirmation d'inscription, ma facture, ainsi que les directives d'accès à l'atelier

FOCUS ON The Product Backlog

Theme	Title (Story)	Type	Est.	Details/ Conditions OK	Priority	Sprint	Comment
Trial	As a potential Fitness club member, i should be able to sign-up as a trial member and print temporary badge so i can try fitness center facilities	feature	L	1) I should be able to enter my name, phone number, and email addres so i can sign up as a trial member 2) The webpage should generate a membership badge with scannable barcode that i could print; 3) The webpage should email a temporary badge ID with scannable bar code 4) The membership should be valid for 5 calendar days from the date of signup 5) The temporary badge should display Trial member	3	1	
Information	As a internet user, i should be able to view the calendar of activities of the fitness club so i can sign up	feature	M	………….	4	2	
Bug	Menu overlap with controls on iPaD tablet with Safari Browser	bug	S				

EPICS & THEMES

EPICs: An EPIC is a large user story. A more complete definition of an epic is that it represents a business work flow or process, and is too large to be estimated.

Example: « As a customer rep in a Mobile Telco agency, i should be able to set up a new line so that one customer can use his mobile phone one hour after leaving the shop »

It includes a complete workflow and must be decomposed in several user stories, maybe several Epics.

Theme: A Group of user stories.

Example:

– All user stories related to a user that has not subscribed a contract but just wants to get information on the fitness club on Internet. Theme= Information

– All user stories related to Trial members: Theme=Trial

– For YAKASSEROLE Theme=recette

AGILE ESTIMATION: Story Points

AGILE ESTIMATION is done by Development team

Initial Estimation is NOT Coding time but estimation of overall effort to convert a Product Backlog Item into a Working Software

Instead of ABSOLUTE estimation, like hours or days,

AGILE ESTIMATION uses RELATIVE ESTIMATION: Task 1 will be 2x longer to implement than Task 2

AGILE ESTIMATION often uses NON-LINEAR SCALES: Uncertainty associated with bigger items is Higher.

– Fibonacci: 1,2,3,5,8,13

– Exponential: 1,2,4,8,16

– T-Shirt sizes: XS-S-M-L-XL

Velocity : The number of story points executed suring 1 Sprint

POKER Planning:

Each Development team member provides its own estimate (card)

Conversation between team member on why will lead to another estimate

Consensus vs majority – is KEY.

SCRUM Events

- SPRINT is the Container event for:
  - 1) SPRINT PLANNING
  - 2) DAILY SCRUM
  - 3) SPRINT REVIEW
  - 4) SPRINT RETROSPECTIVE

KANBAN

– From LEAN/JIT manufacturing TOYOTA: Reduce inventory when producing cars in the 50s

– KAN=Visual BAN=Card

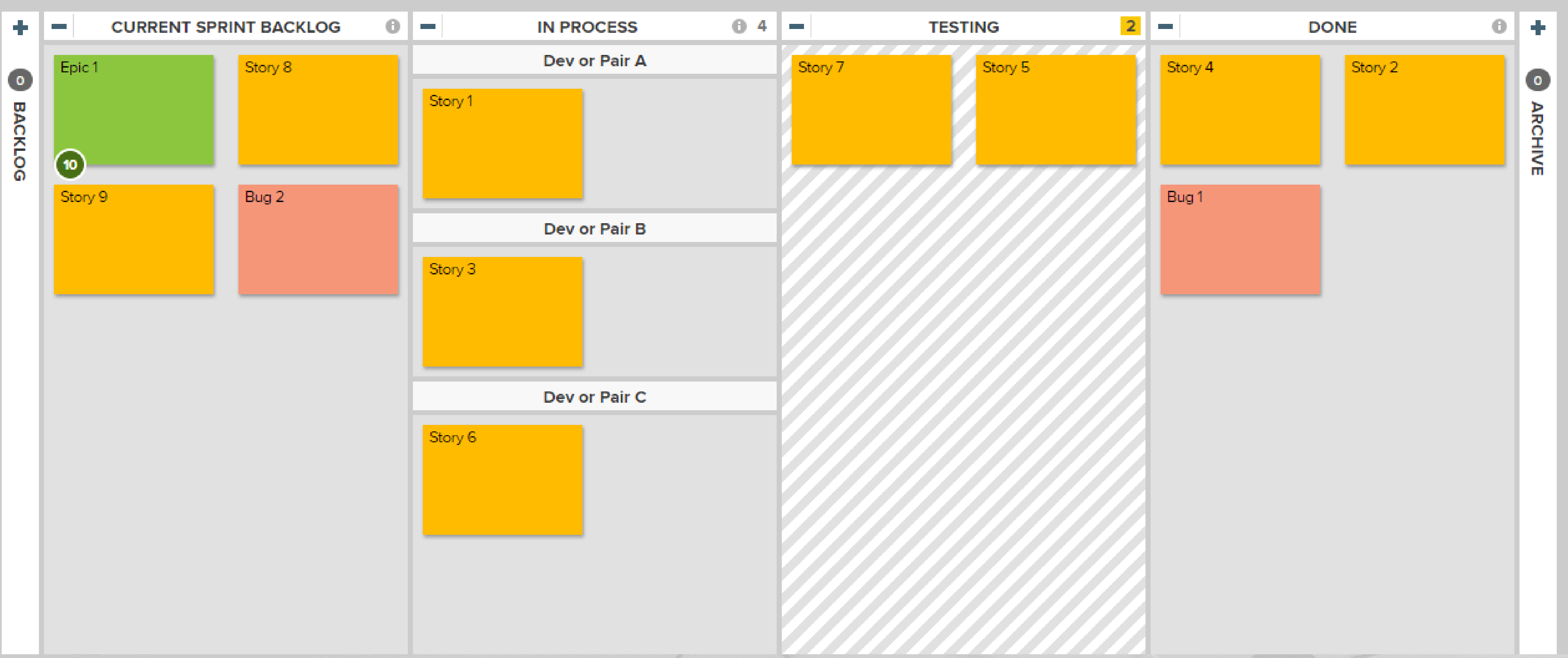
– Visualy appealing, and efficient way to process work items

– PULL system (While SCRUM considered to be PUSH)

– KANBAN Board – Used in conjuction with SCRUM (Example LeanKit)

– WIP : Work In Progress – Limit the number of tasks in //

KANBAN BOARD



BACKLOG 0

CURRENT SPRINT BACKLOG
- Epic 1 (10)
- Story 8
- Story 9
- Bug 2

IN PROCESS 4
- Dev or Pair A
  - Story 1
- Dev or Pair B
  - Story 3
- Dev or Pair C
  - Story 6

TESTING 2
- Story 7
- Story 5

DONE
- Story 4
- Story 2
- Bug 1

ARCHIVE 0

XP: eXtreme Programming

Collection of SW engineering practices (1996) : More prescriptive and implementation centric –

Used in combination with SCRUM

– Weekly iterations, contained in a Quarterly cycle.

– Split Stories into Tasks,

– Cross Functionnal team – include customer

– Colocated Team /Physical charts – Posters – Electronic dashboards

KEY Development Techniques:

– Test Driven Development: is thinking about test scenarios before implementation

– Refactoring: is an activity that improves maintainability of code without changing the code's external behavior.

– Incremental Design.

– Pair Programming: 2 developers sit at a computer terminal, one writes code while the other views the monitor. The second person constantly reviews the code as it is being developed, asks questions about the implementation, and assists the first developer with code suggestions. The two developers switch roles periodically and the process continues.

AGILE & DevOps Complement each other

AGILE Unifies Business-Development and Testing (Q&A) Organizations in a single Process.

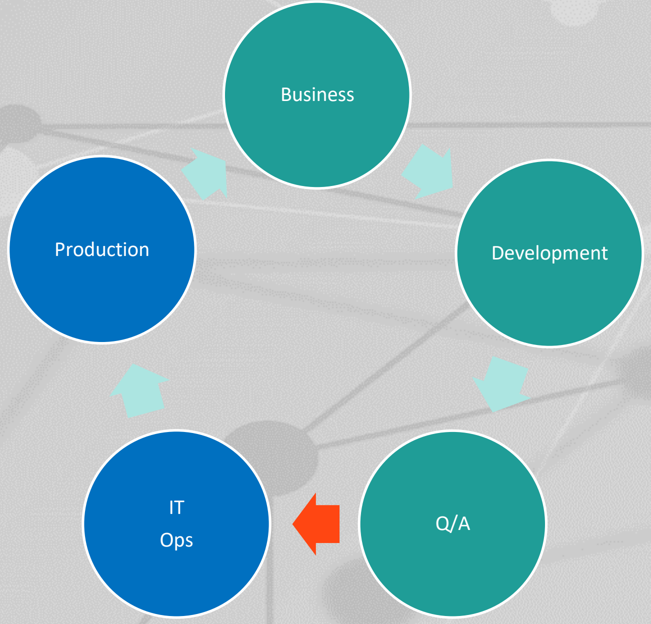
But Rapid Cadence of New increments (versions) becomes a Nightmare for IT Operations.

DevOps brings 3 key elements and a change of mindset to ENABLE AGILE:

1) CI Continuous Integration: is the practice of frequently committing changes to source code

2) CD Continuous Delivery:is the capability to always keep a product in a stable state, after every change, so that the product is always potentially deployable

3) Continouous Deployment: means automatically deploying the product increment to production. Continuous deployment may or may not follow continuous delivery.



SCALING AGILE

Applying SCRUM in a multi-team context:

– NEXUS: SCRUM add-on - Free online GUIDE and poster: www.scrum.org

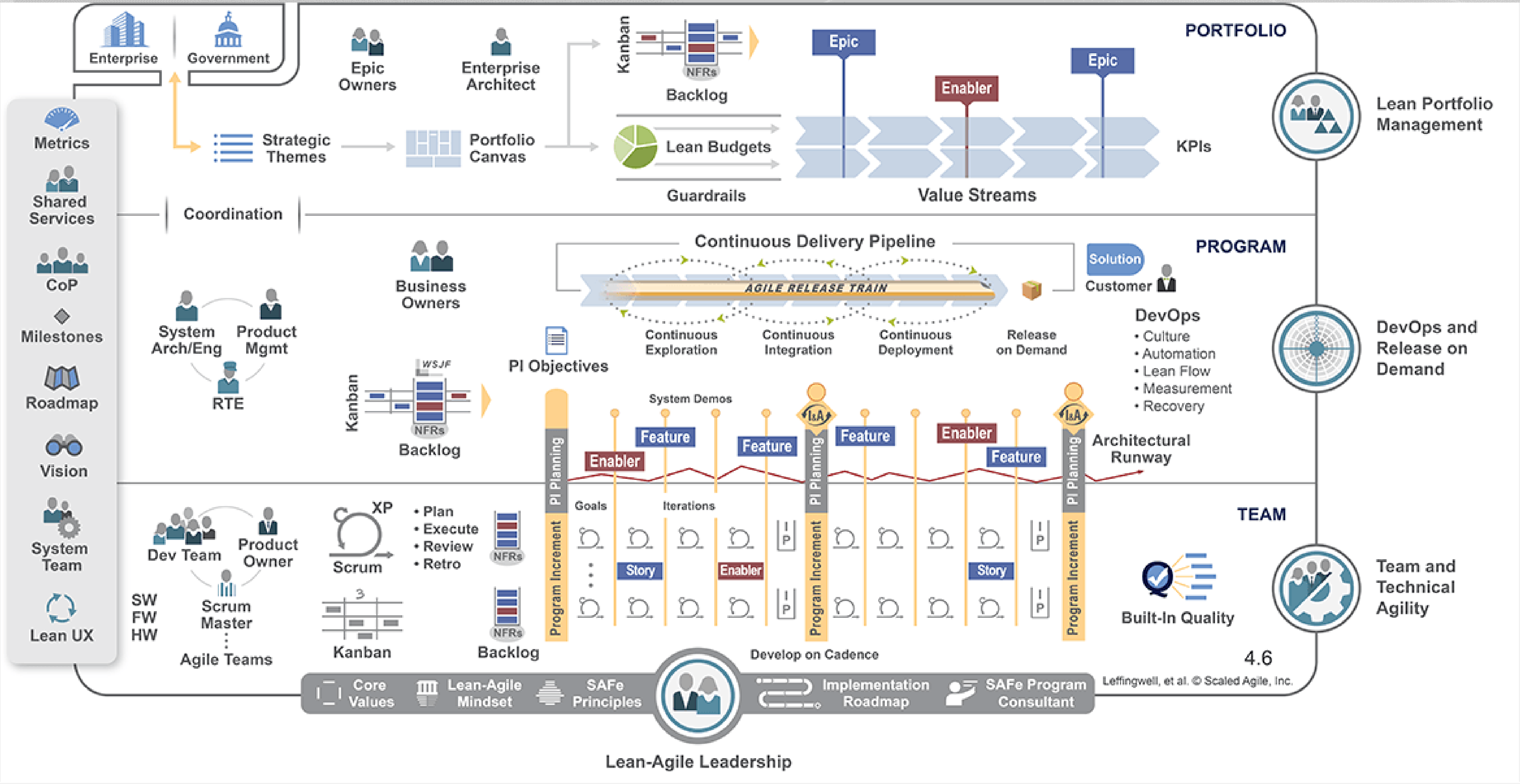
– Scrum@Scale : https://www.scrumatscale.com/scrum-at-scale-guide/ S@S Guide

– LeSS or « Large Scale SCRUM »: https://less.works/less/principles/large_scale_scrum_is_scrum.html

FRAMEWORK:

– SAFe 4.6 (since 2011): https://www.scaledagileframework.com/

SAFe: Scaled Agile Framework



EPITA

ÉCOLE D'INGÉNIEURS EN INFORMATIQUE