Agile Introduction

Ingegneria del software UNIPI

davideluca.roitero@gmail.com

Marzo 2019

Team Building

Davide Roitero

56 anni ben portati, informatico da circa 42.

Ho conosciuto Agile nel 2010, ne sono rimasto affascinato e, da allora, non ci siamo più separati.

Sono Agile Coach in una grande azienda e seguo la componente metodologica dei progetti.

Nel mio tempo libero, per rilassarmi, produco ottimi gorgonzola con un approccio iterativo ed incrementale che cerca la perfezione.

Perché sono qui oggi

Potremmo limitarci a imparare qualcosa, e già non sarebbe poco, ma io, da quando lavoro con Agile, sono più protagonista delle mie giornate e più contento di alzarmi al mattino per andare a lavorare.

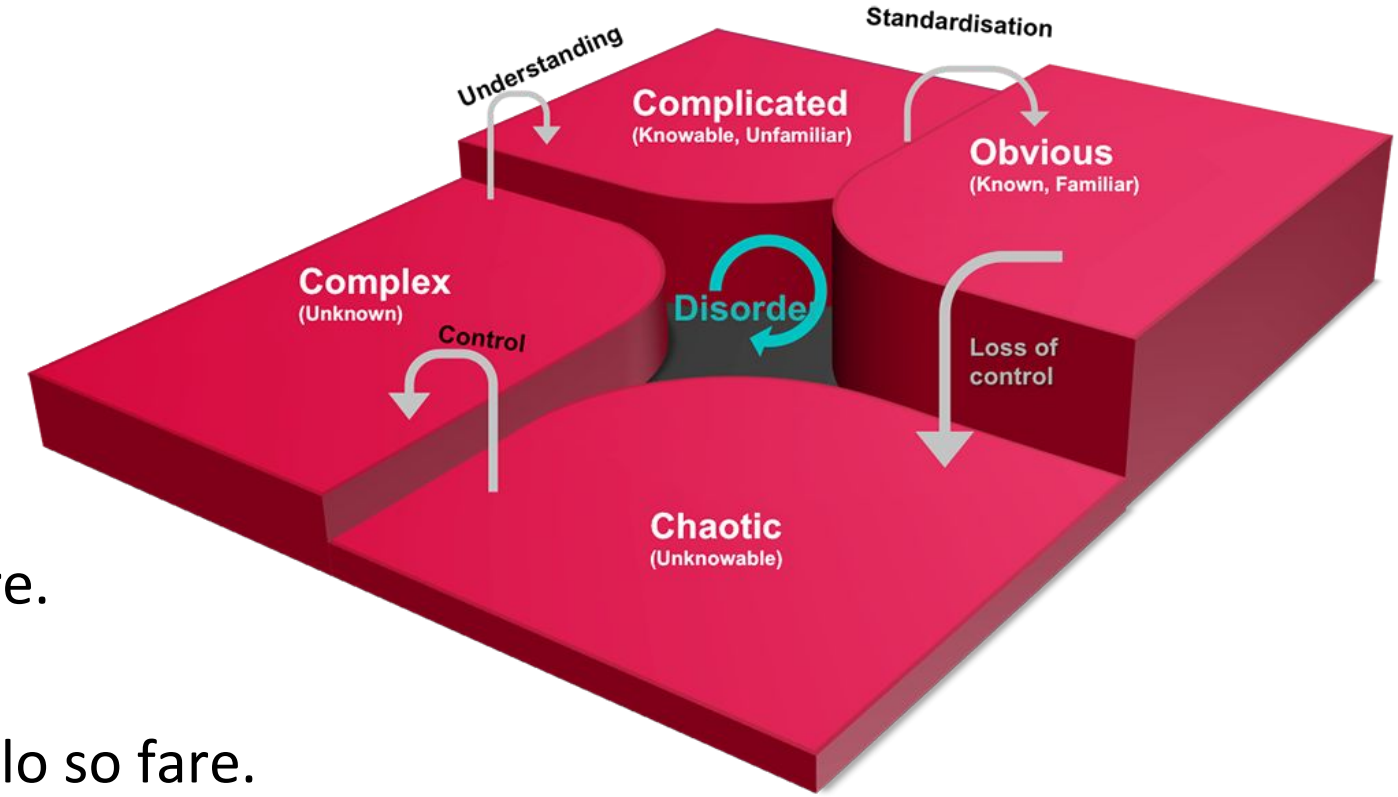
Vorrei che anche per voi fosse lo stesso.

E’ un obiettivo ambizioso? Probabilmente sì, ma mi pare un’ottima ragione per essere qui oggi.

Se ci sarò riuscito me lo direte tra un bel po’ di tempo.

La mia email è qui sotto.

Cynefin Framework



Ovvio: so cosa c'è da fare. Lo so fare.

Lo faccio

Complicato: so cosa c'è da fare. Non lo so fare.

Chiamo un esperto.

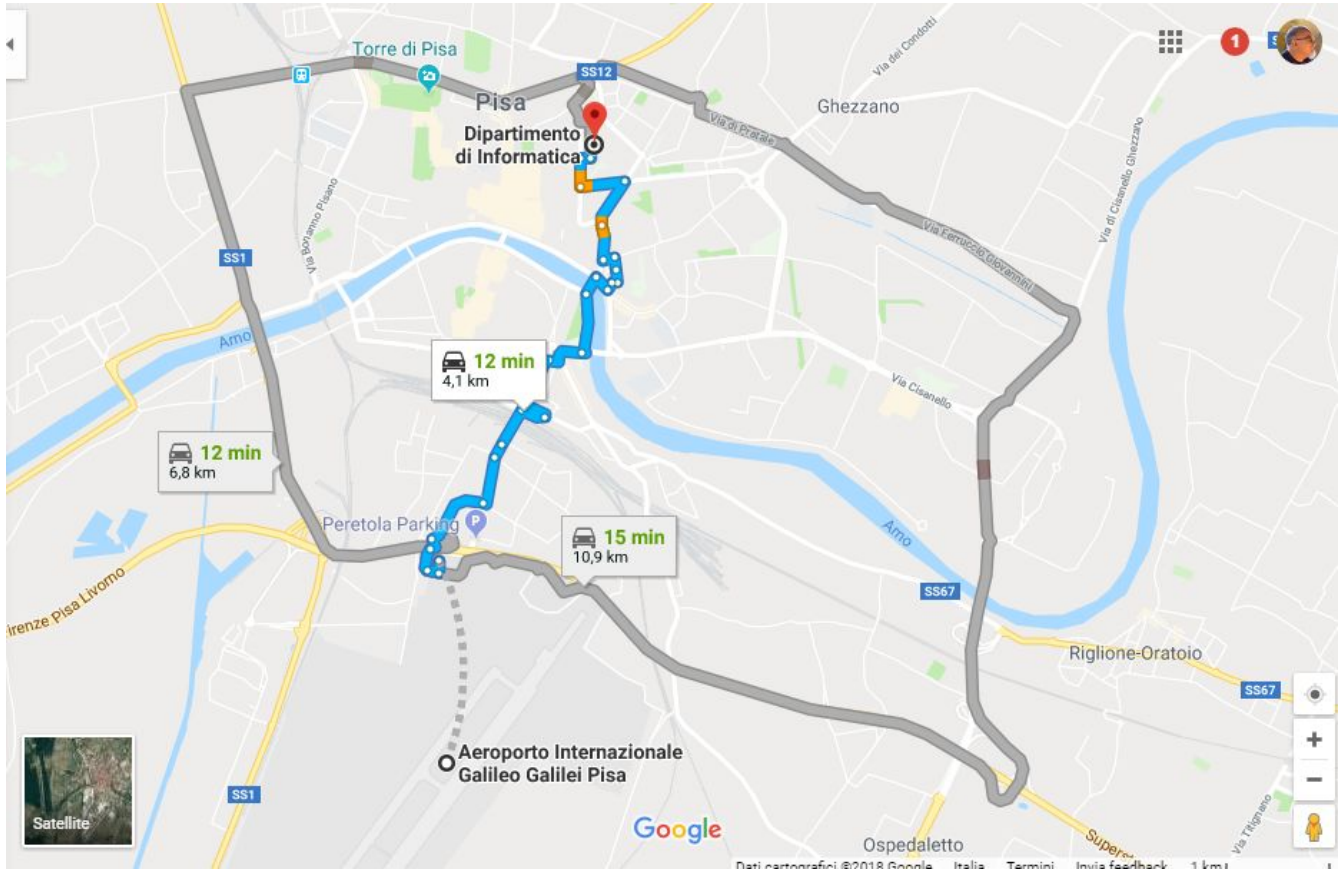
Complesso: non so cosa c'è da fare e non lo sa nessuno.

Posso fare degli esperimenti e imparare per migliorare.

Caotico: Non si capisce nulla.

Devo fare la prima cosa che posso fare e poi vedere cosa succede.

Noi conviviamo con la complessità

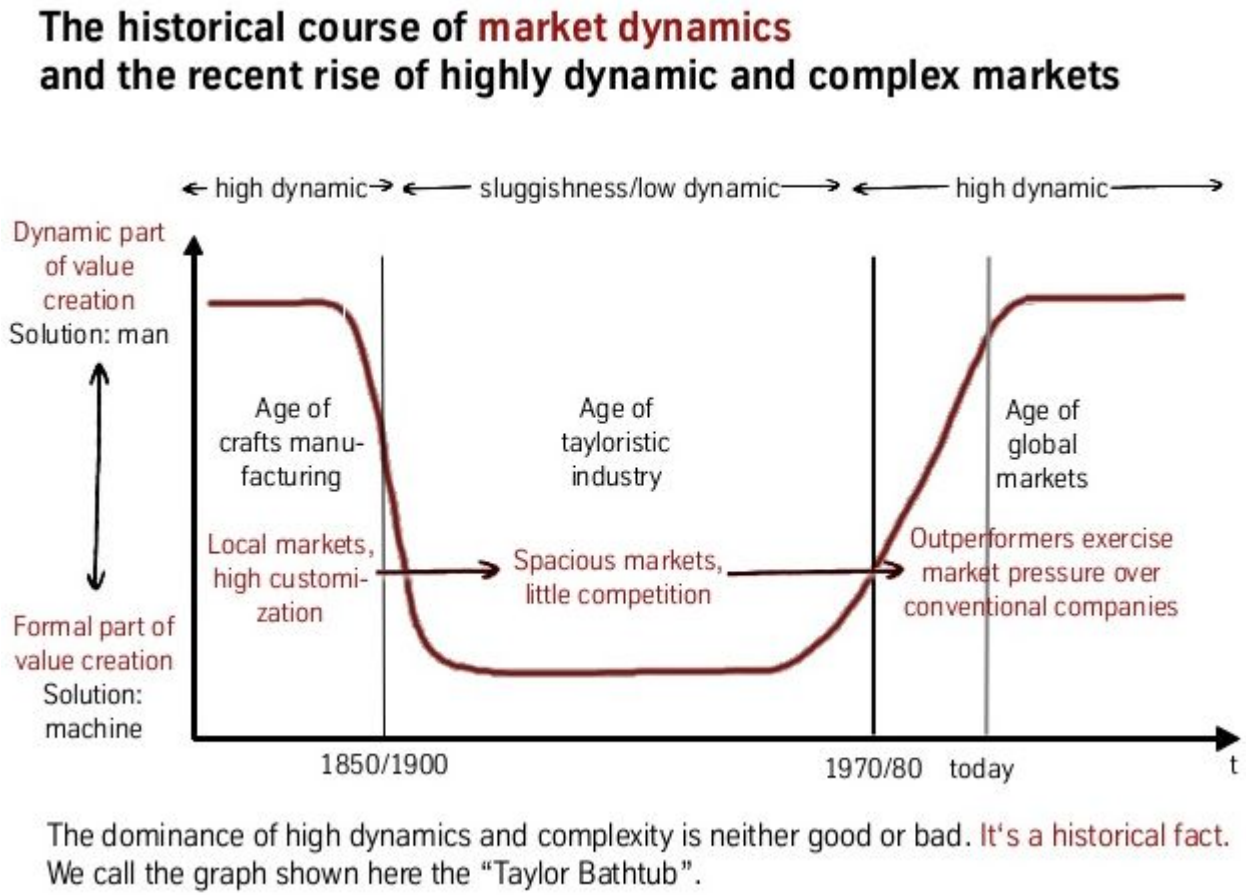


Se devo venire qui dall'aeroporto, in macchina, imposto il navigatore ed inizio a guidare.

Se a metà strada incontro un ingorgo cambio strada e lui ricalcola il percorso tenendo fissa la meta così come se si risolve un ingorgo mi fa cambiare strada.

Il navigatore è molto Agile!

Taylor bathtub



Nel mercato odierno i Doer e i Thinker sempre più spesso coincidono.

Il mercato è globale.

Abbiamo a disposizione tecnologie a basso costo che ci permettono prototipazioni rapidissime.

Possiamo raccogliere facilmente feedback dai nostri customer.

Parliamo di Agile al fine di ….

«Al fine di…» sono soltanto tre parole, ma a me hanno cambiato modo di pensare e lavorare.

Agile ci chiede di pensare sempre per chi e perché sviluppiamo software e spesso questo ci aiuta a trovare la strada migliore.

Parliamo di Agile per conoscere un altro modo di fare progetti, per avere nella nostra cassetta qualche attrezzo in più da usare all'occorrenza.

Ne parliamo anche perché viviamo in un mondo veloce e complesso e Agile è particolarmente a suo agio in questo contesto.

Agile Ways of Work

Agile nasce da Lean Thinking

Il Lean Thinking (Pensare Snello) è una strategia operativa nata dal mondo automotive, in Toyota, ma oggi universalmente applicata in settori e ambiti diversi per aumentare l'efficienza ed eliminare gli sprechi.

È una strategia operativa perché racchiude, insieme all'inquadramento sul pensiero e sulle teorie organizzative, anche l'approccio pratico (il lavoro umano che serve per realizzare la conversione snella).

I principi su cui si basa il Lean Thinking sono semplici: il punto di partenza è l'identificazione degli sprechi, da cui ci si muove per poi eliminarli e produrre di più con un minor consumo di risorse.

Lean Thinking

Lean Thinking è una filosofia produttiva che ricerca la perfezione.

Lean Thinking è un sistema di lavoro che cerca di ridurre il tempo tra l'idea e la vendita

I principi del Lean Thinking

Identificare ciò che vale (value)

Individuare ciò per cui i clienti sono disposti a pagare un prezzo

Identificare il flusso del valore (value stream)

Allineare le attività che creano valore nella giusta sequenza

Far scorrere il flusso del valore (flow)

Mettere in atto le attività a valore senza interruzioni

Fare in modo che il flusso sia tirato (pull)

Fare scorrere il flusso in base alle richieste del cliente

Puntare alla perfezione (perfection)

Assumere la perfezione come riferimento per programmi di miglioramento continuo

Estendere alla catena dei fornitori (lean supply chain)

Affrontare anche la catena dei fornitori in ottica lean

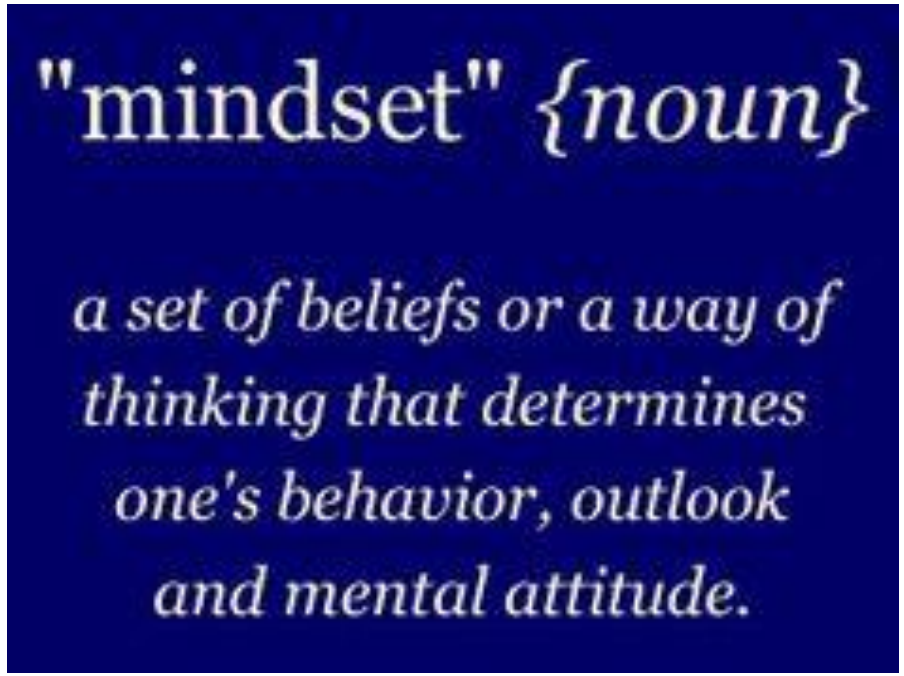
I cambiamenti generati dal Lean Thinking

L'applicazione in azienda dei sei principi provoca un grande cambiamento sia sul piano "fisico" sia sul piano organizzativo (riduzione dei livelli gerarchici, orientamento ai processi, team interfunzionali, responsabilizzazione, delega e sviluppo competenze a livelli operativi, snellimento delle funzioni ecc.).

Tutto questo comporta quindi un radicale cambiamento di mentalità da parte di tutto il personale ed una vera e propria "rivoluzione culturale".

È una scelta coraggiosa, operata da una direzione forte, dinamica, innovativa e moderna

Agile è un mindset



Agile è un mindset che trae ispirazione dai principi del Lean Thinking adattandoli per lo sviluppo del sw.

I principi a cui si ispira Agile sono:

- MURI (Pull)
- MURA (Flow)
- MUDA (Value)
- KAY ZEN (Perfection)

Muri

Muri è lo spreco legato alla sovraproduzione di pezzi.

Oltre alle risorse immobilizzate fino all'utilizzo, se intervengono cambiamenti di mercato l'azienda non può adeguarsi fino a che le scorte non sono finite.

Sviluppo della UI di tutto il progetto a priori

Mura

Mura è lo spreco legato alla incostanza della produzione.

Se ogni giorno parte un container che può trasportare due auto, quando ne produco una parte semivuoto e quando ne produco tre me ne rimane una nel piazzale.

Nel sw si alternano deliverable di dimensioni molto diverse tra loro

Muda

Muda sono gli sprechi.

Gli sprechi non sono solo i prodotti che devo buttare ma anche tutte le attività del processo produttivo per cui il cliente non è disposto a pagare, a cui non riconosce un valore.

Trasporti, attività non necessarie, movimentazioni, difetti, inventari e attese

Kai Zen

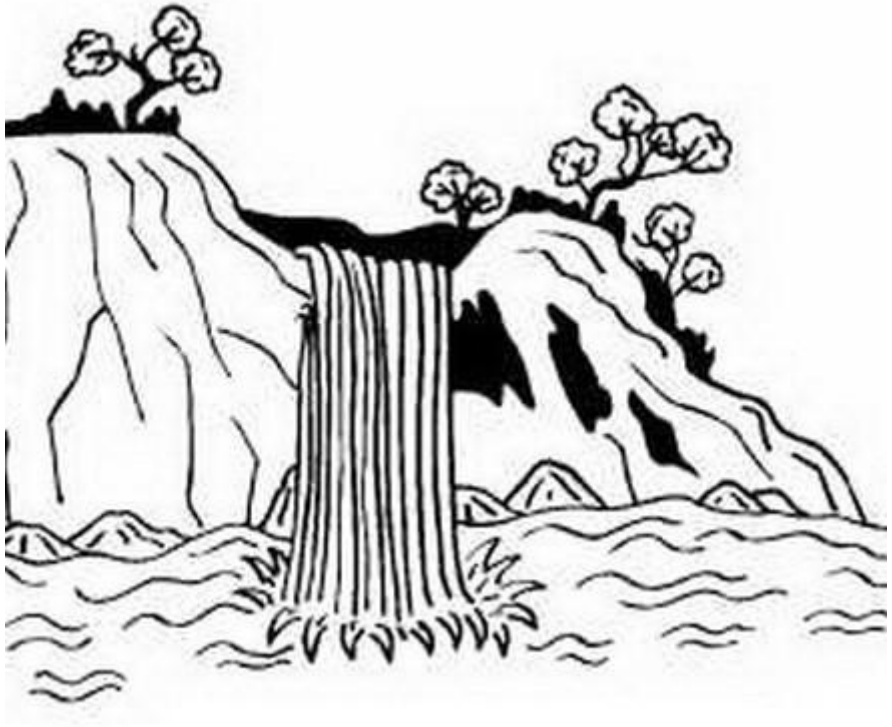
Il Kai Zen è il miglioramento continuo.

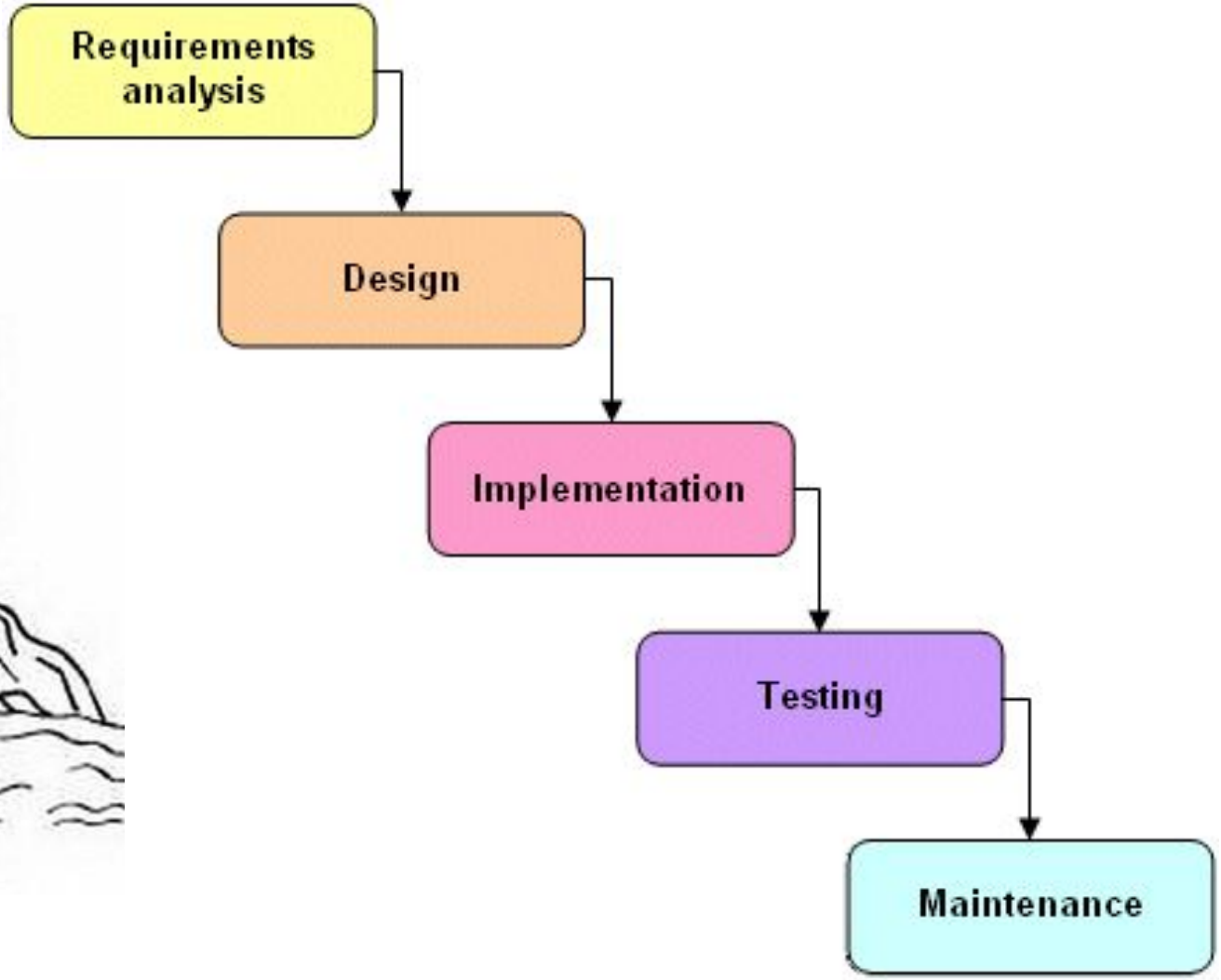
Tutto il team di produzione è incentivato a pensare a come migliorare la produzione.

Il momento migliore per perfezionare il processo produttivo è quando sta funzionando.

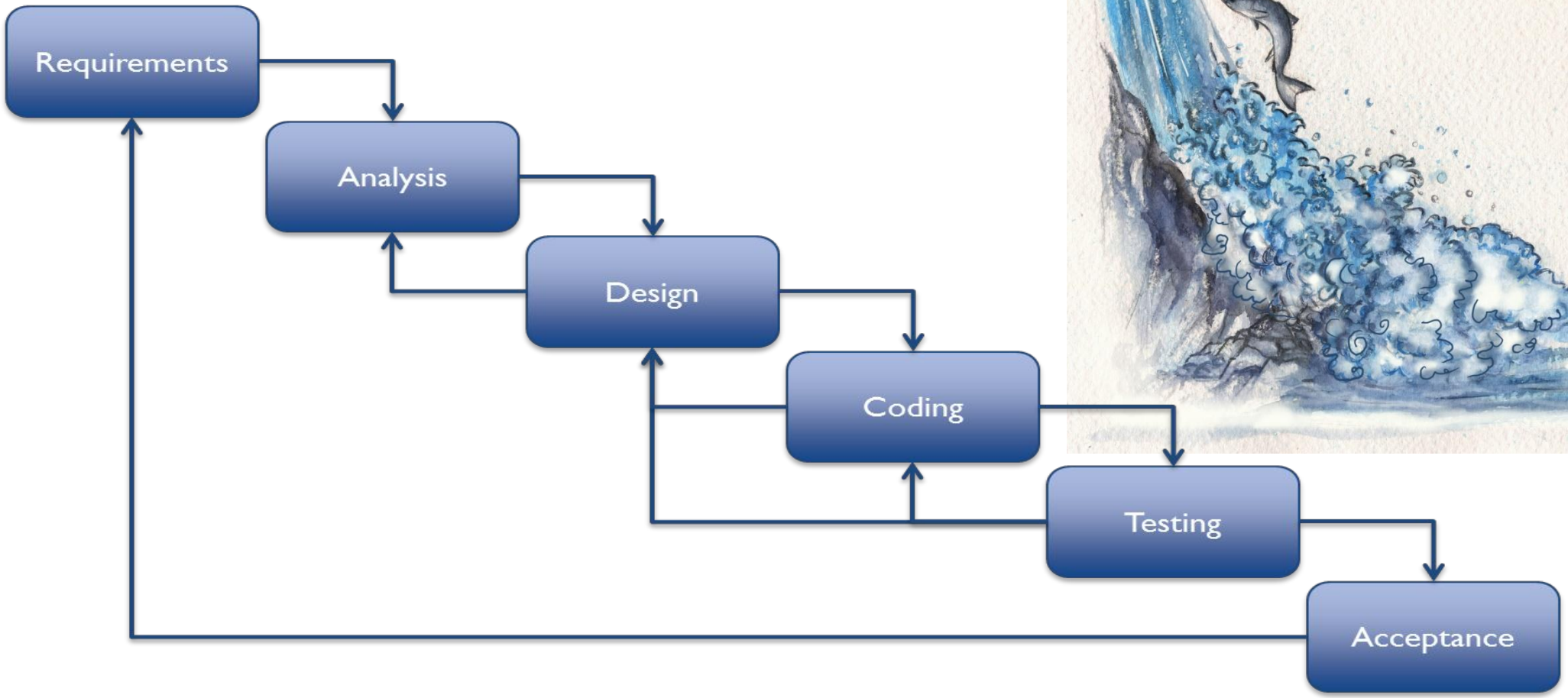
Quando non funziona vorranno solo che lo aggiustiamo in fretta.

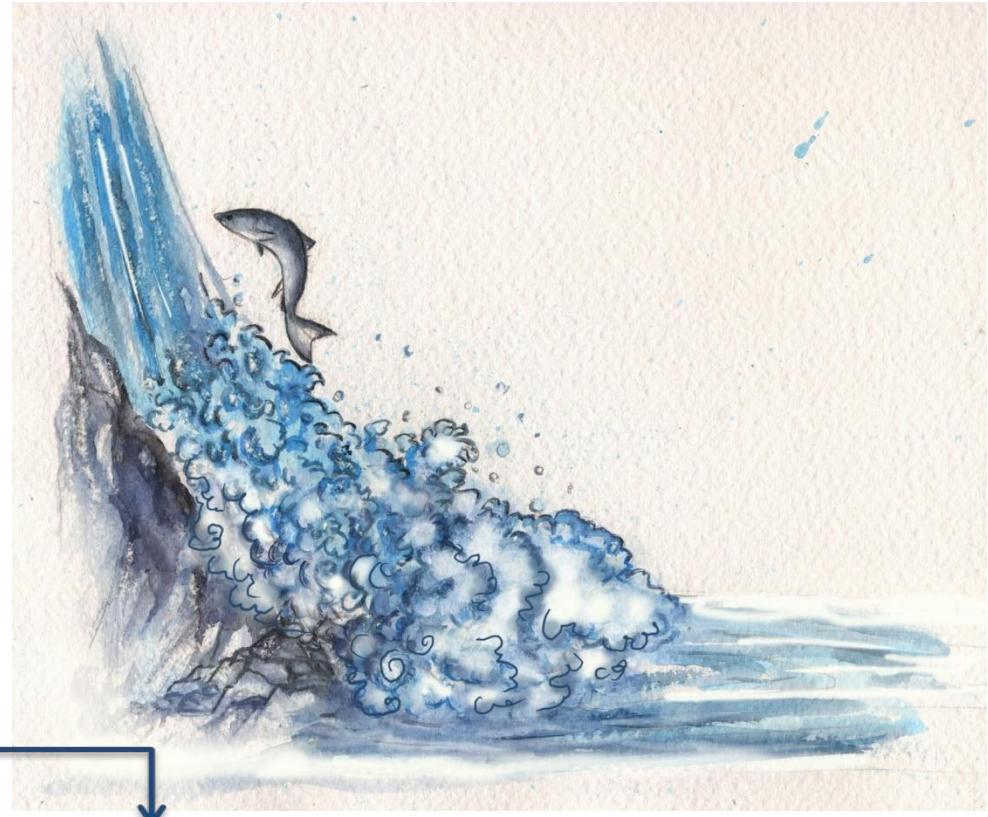
I progetti waterfall nella teoria



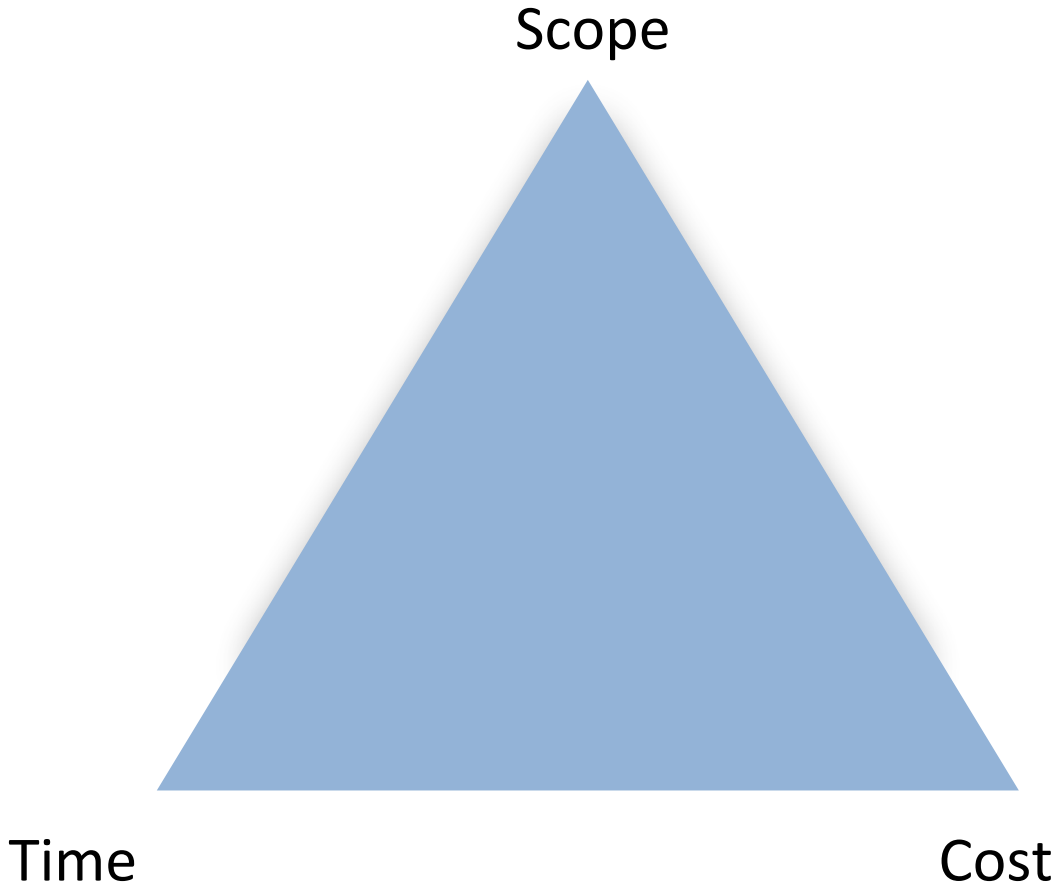


I progetti waterfall nella pratica





I progetti waterfall che finiscono bene

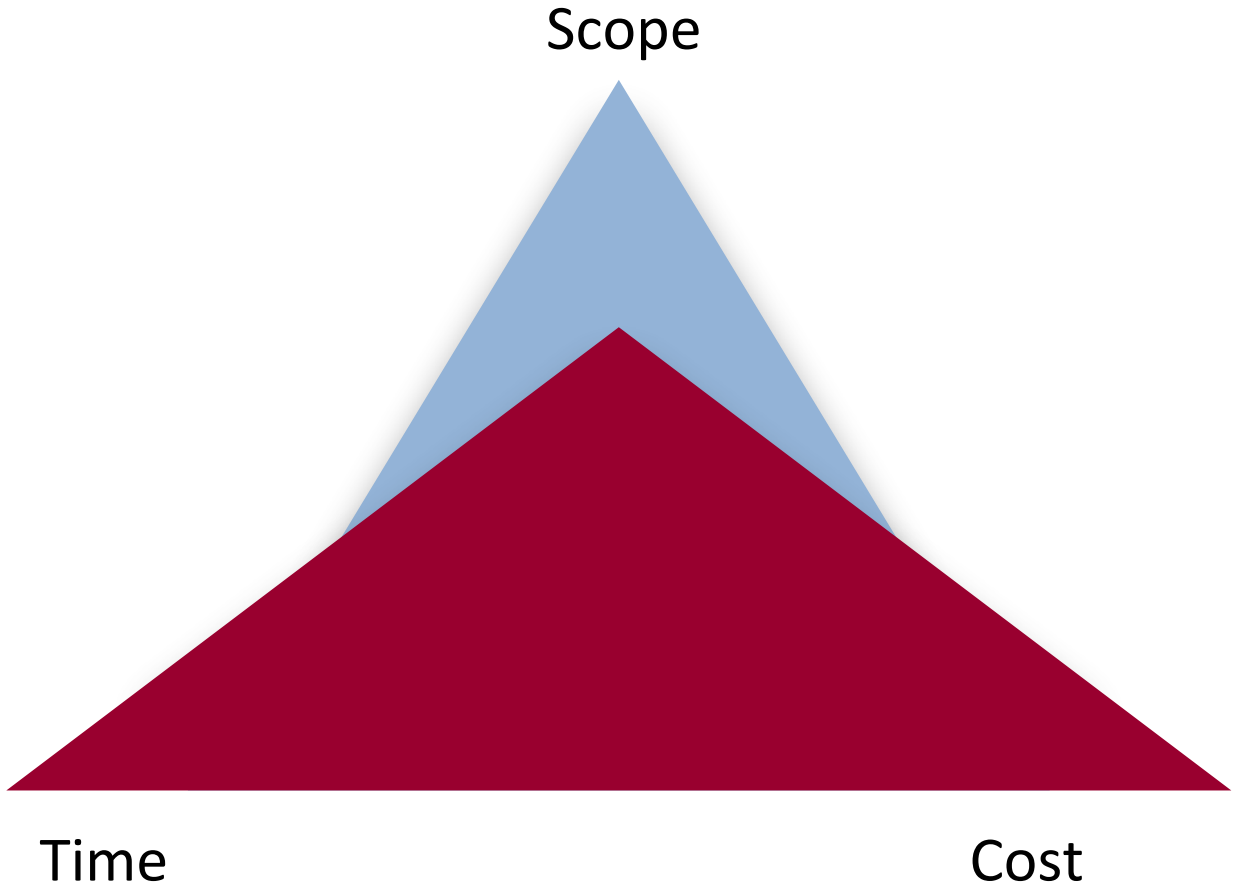


Nei progetti waterfall mantengo fissi tempi, costi e funzionalità del progetto.

Se sono un bravissimo capo progetto riesco a fare nel tempo stabilito e con i soldi pattuiti le funzionalità richieste.

Peccato che è passato un anno e i clienti che vogliono quello che ti avevo chiesto non ci sono più.

I progetti waterfall che finiscono meno bene

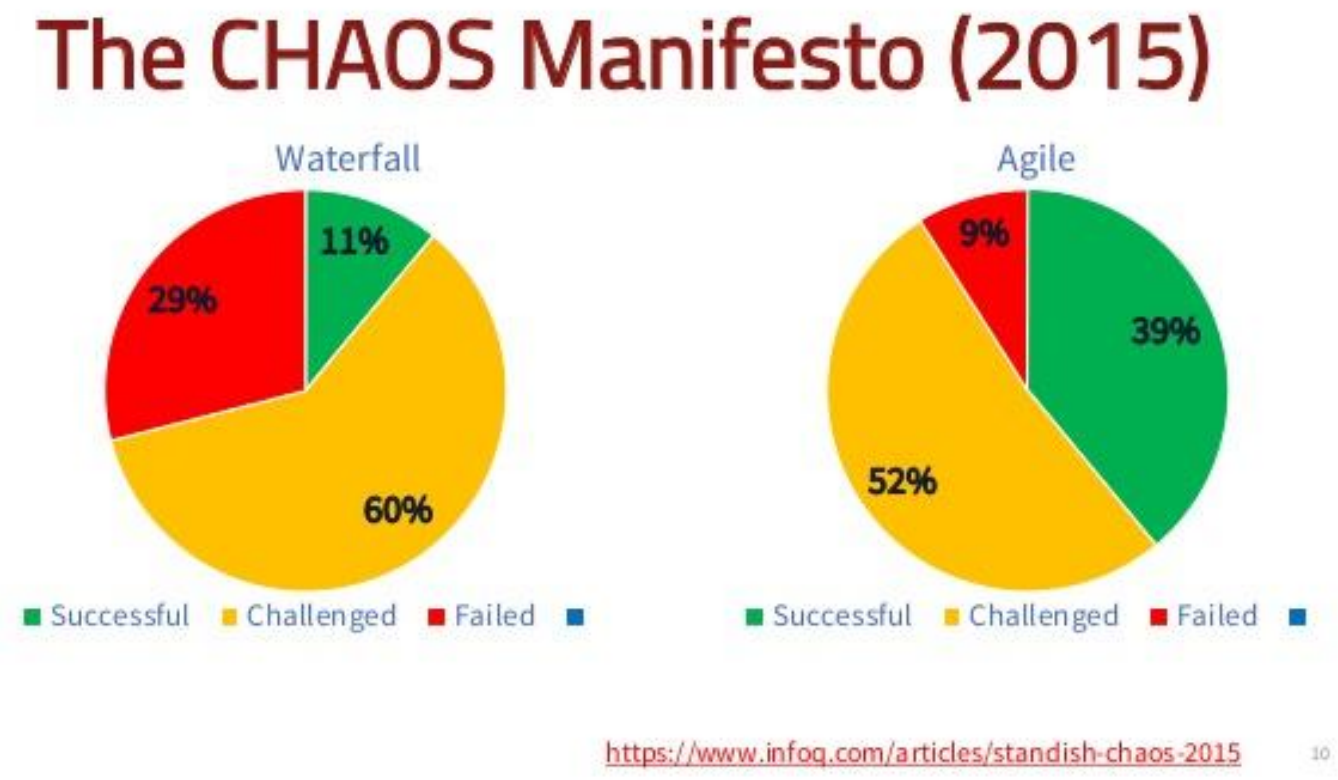


Quasi sempre:

- Non finisco in tempo
- I costi aumentano
- Le funzionalità non ci sono tutte

Secondo voi che percentuale di progetti finisce male?

Chaos Manifesto

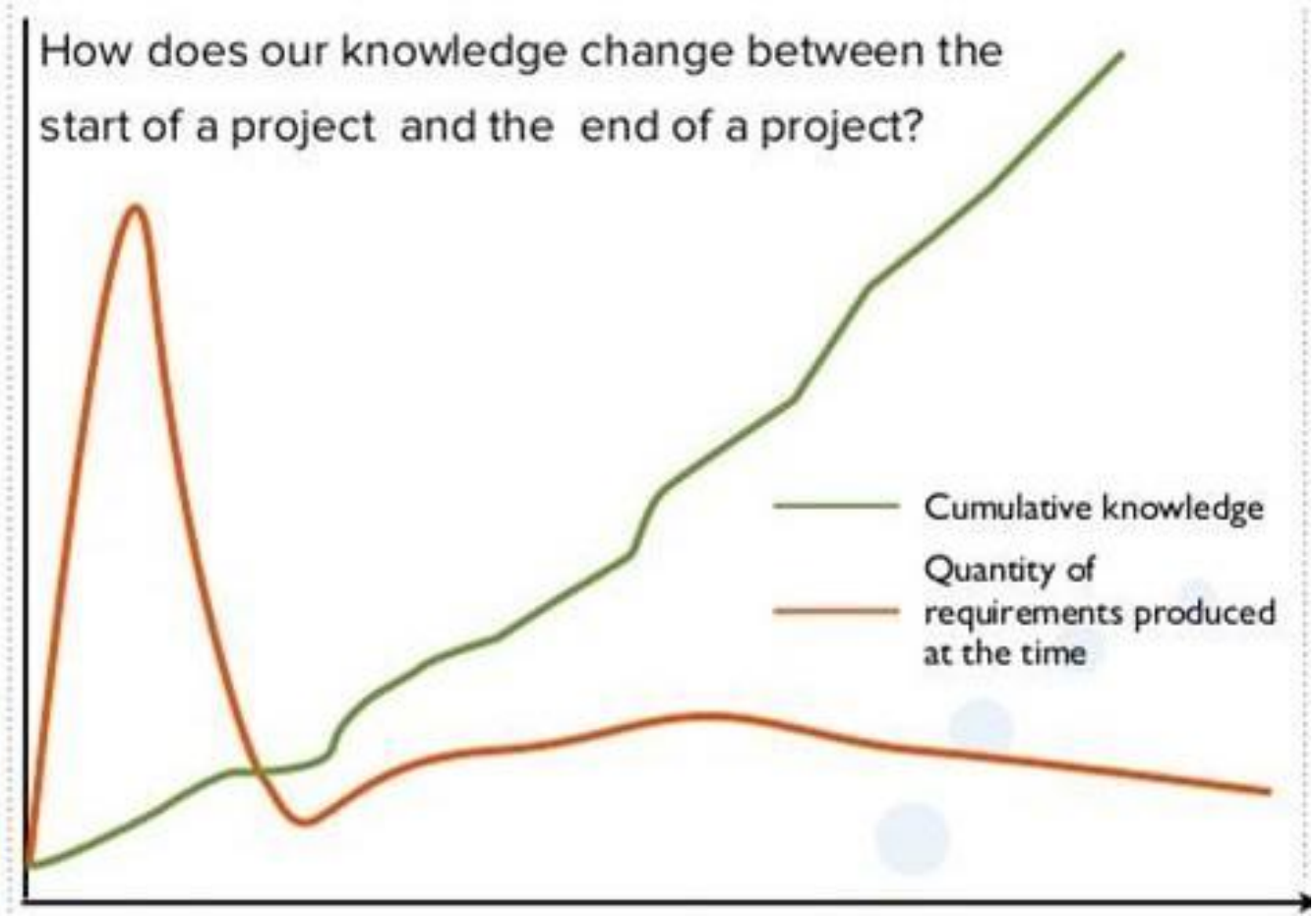


Agile non è la soluzione di tutti i problemi.

Anche Agile può essere applicato male e ce ne sono esempi ovunque.

Di certo, con Agile diminuisce drasticamente la percentuale di progetti che falliscono e aumenta drasticamente la percentuale di quelli che finiscono on time, on budget, on scope.

Requisiti vs Conoscenza



«In principio erano i requirements»

Ma il cliente, all'inizio non sa cosa vuole.

Tantomeno sa cosa vogliono i suoi clienti.

Lo scopre vedendo e lo scopre vendendo.

Minimum Valuable Product

Devo realizzare la minima porzione di prodotto che mi dia modo di raccogliere i feedback dei miei consumatori per capire se sto sviluppando quello che serve loro.

L'MVP di Dropbox era un filmato di due minuti, realizzato in una mattinata, fotografando i disegni di un membro del team.

AGILE manifesto

Gli individui e le iterazioni	più che	i processi e gli strumenti
Il software funzionante	più che	la documentazione esaustiva
La collaborazione con il cliente	più che	la negoziazione dei contratti
Rispondere al cambiamento	più che	seguire un piano

Ovvero, fermo restando il valore delle voci di destra, consideriamo più importanti le voci a sinistra

AGILE principle #1

La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua

AGILE principle #2

Accogliamo i cambiamenti nei requisiti,
anche a stadi avanzati dello sviluppo.
I processi AGILI sfruttano il cambiamento
a favore del vantaggio competitivo del
cliente

AGILE principle #3

Consegnamo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.

AGILE principle #4

Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.

AGILE principle #5

Fondiamo i progetti su individui motivati.
Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.

AGILE principle #6

Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.

AGILE principle #7

Il software funzionante è il principale metro di misura di progresso

AGILE principle #8

I processi agili promuovono uno sviluppo sostenibile.
Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante

AGILE principle #9

La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità

AGILE principle #10

La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale

AGILE principle #11

Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano

AGILE principle #12

A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza

Agile frameworks

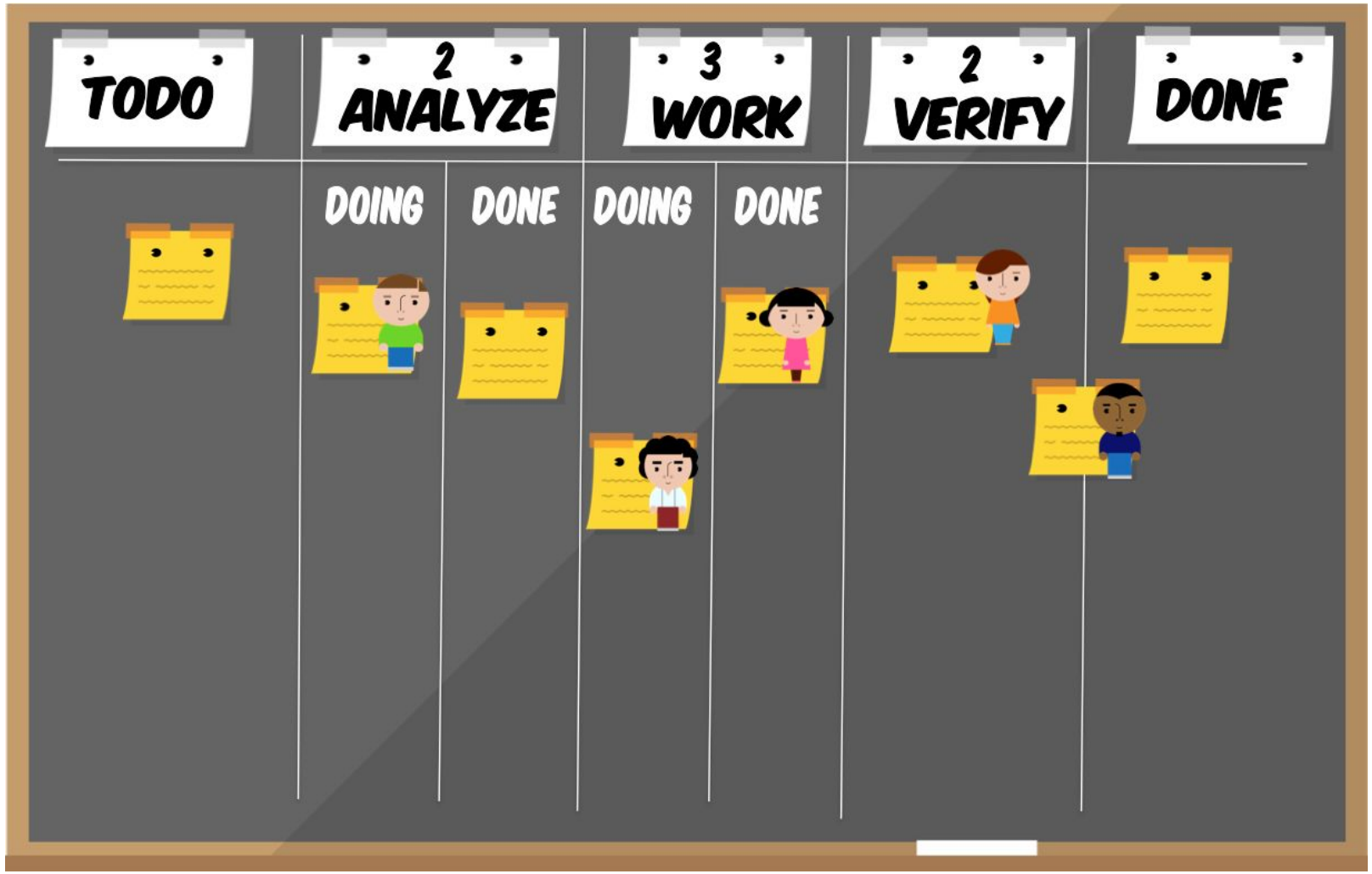
Per un team

- Scrum
- Kanban
- Extreme Programming
- …

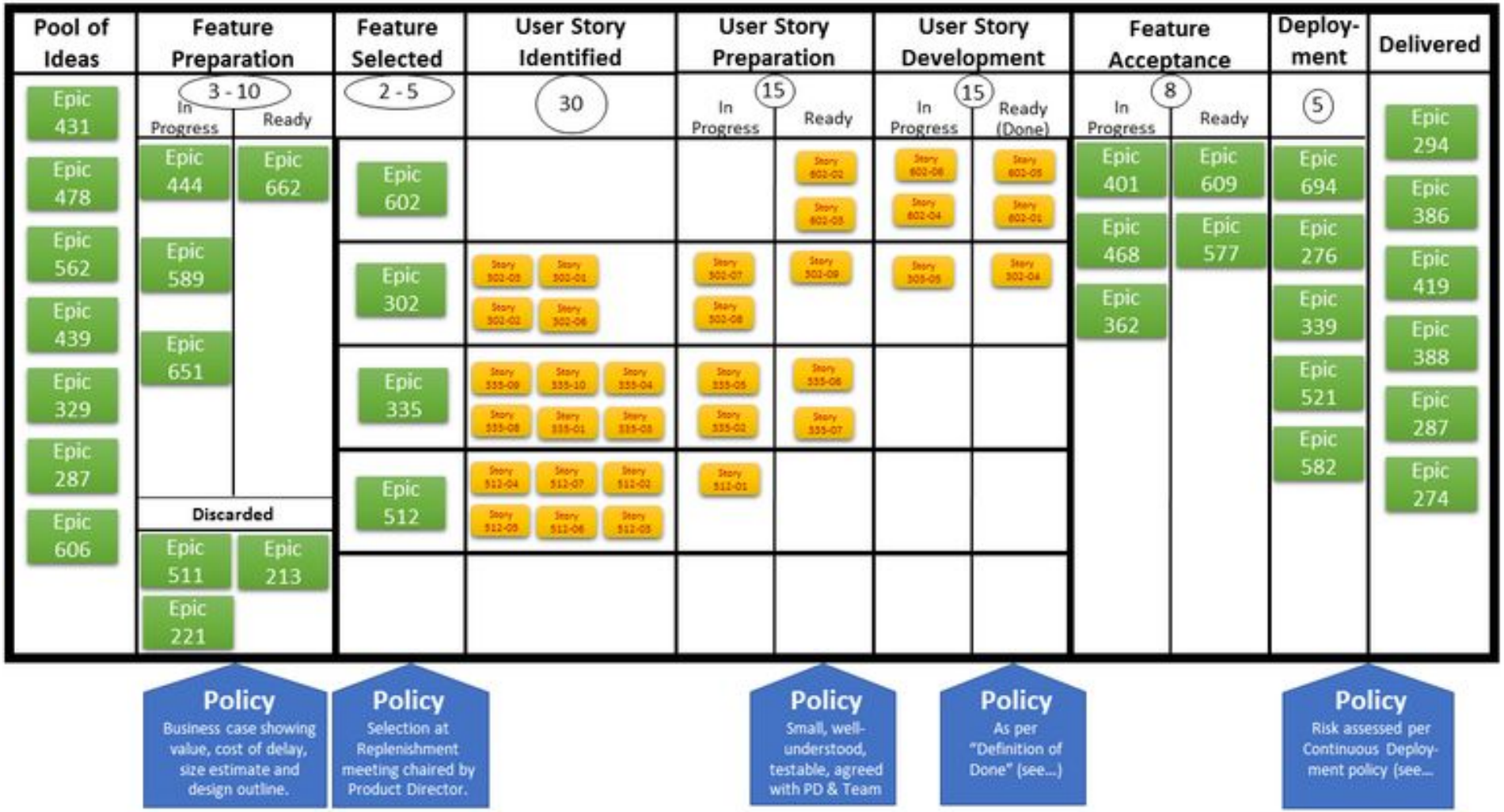
Per più team

- Less
- SAFe
- DAD
- Scrum@Scale
- Nexus
- …

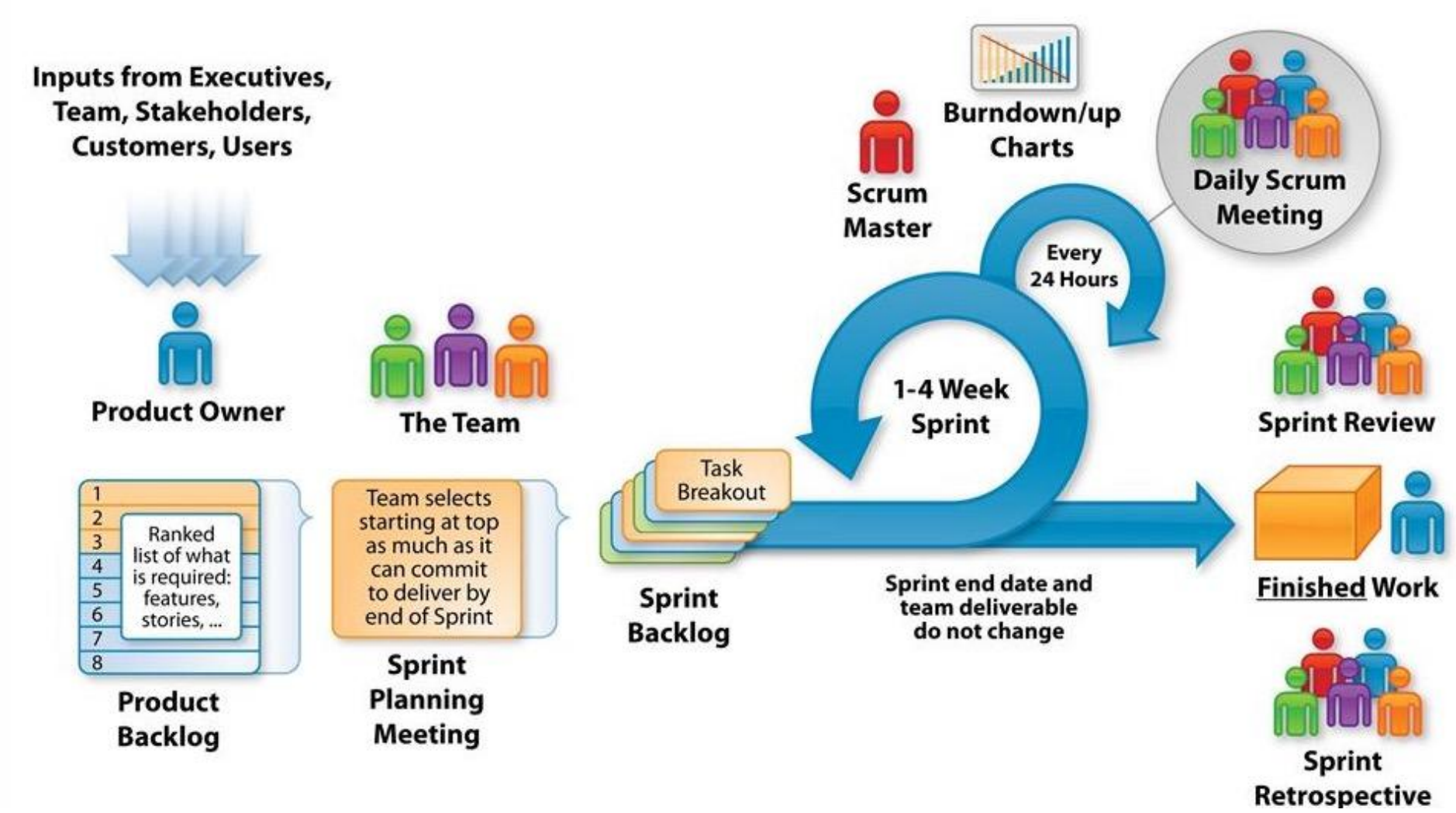
Kanban semplice



Kanban maturo



Scrum


Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting

Scrum Team crossfunzionale

Uno Scrum Team crossfunzionale è un team dove ci sono membri con skill molto specifici che, uniti insieme, possono ottenere il massimo risultato. Nel tempo ognuno di loro allarga il suo know how attraverso la condivisione delle conoscenze.

Nel Team devono essere presenti tutti gli skills necessari a portare a casa il risultato richiesto.

From Concept to Cash.

Scrum Team auto organizzato

Uno Scrum Team riceve dal P.O. le specifiche so COSA deve essere fatto, ma decide in modo autonomo COME realizzare le specifiche.

Al suo interno divide le attività in piccoli task.

Ogni membro prende in carico il task a più alta priorità su cui può lavorare.

Al Daily Meeting ci si allinea e ci si confronta.

Il risultato, alla fine dello Sprint è del team e mai del singolo.

Lo Scrum Master

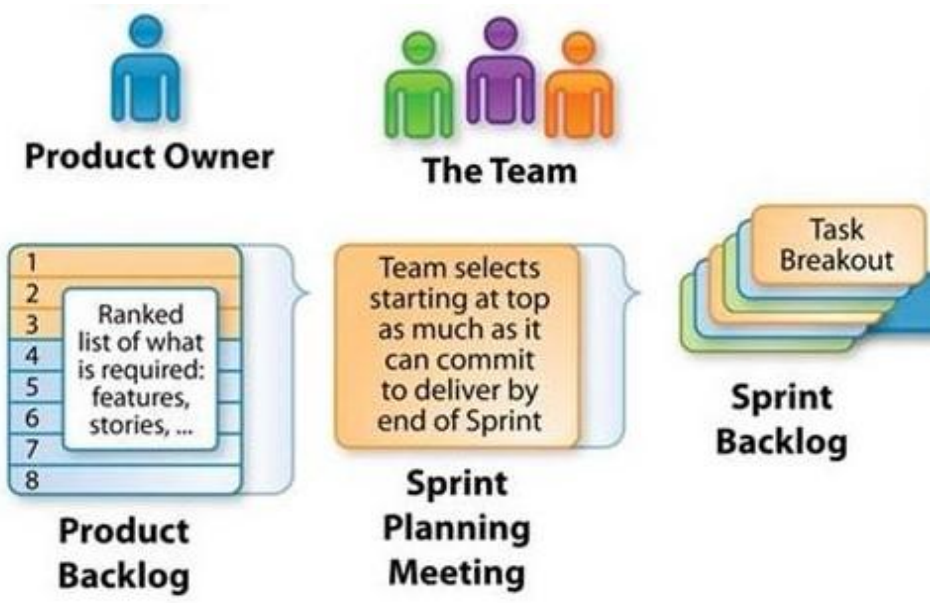
- Aiuta lo Scrum Team ad aderire a Scrum
- Aiuta il Product Owner a scrivere le storie
- Aiuta lo Scrum Team a risolvere gli Impediments

Il Product Owner

Il Product Owner raccoglie input da chiunque: clienti, manager, team ma è l'unico abilitato ad inserire storie nel backlog e a prioritizzarlo massimizzando il R.O.I.

Il team NON RICEVE RICHIESTE DA ALTRI che non siano il Product Owner e solo attraverso il Product Backlog

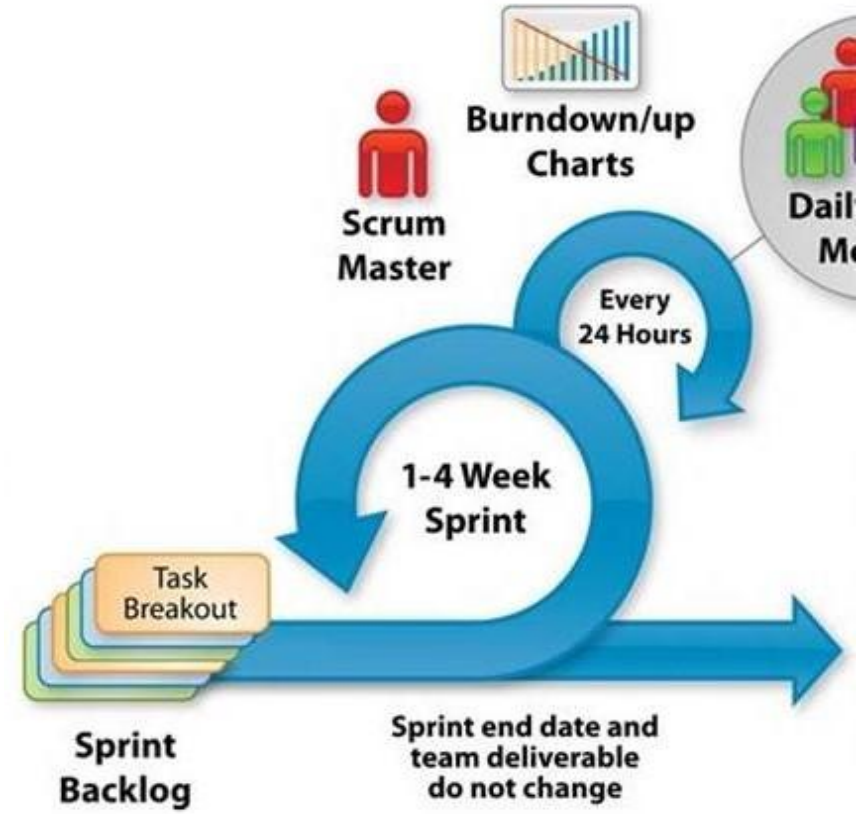
Lo Sprint Planning



Il Product Owner, ad ogni inizio Sprint, propone al Team le storie partendo dalla più alta nel backlog.

Il Team le accetta, le pesa e le inserisce nello Sprint Backlog fino ad esaurire la sua capacità per quello sprint.

Lo Sprint

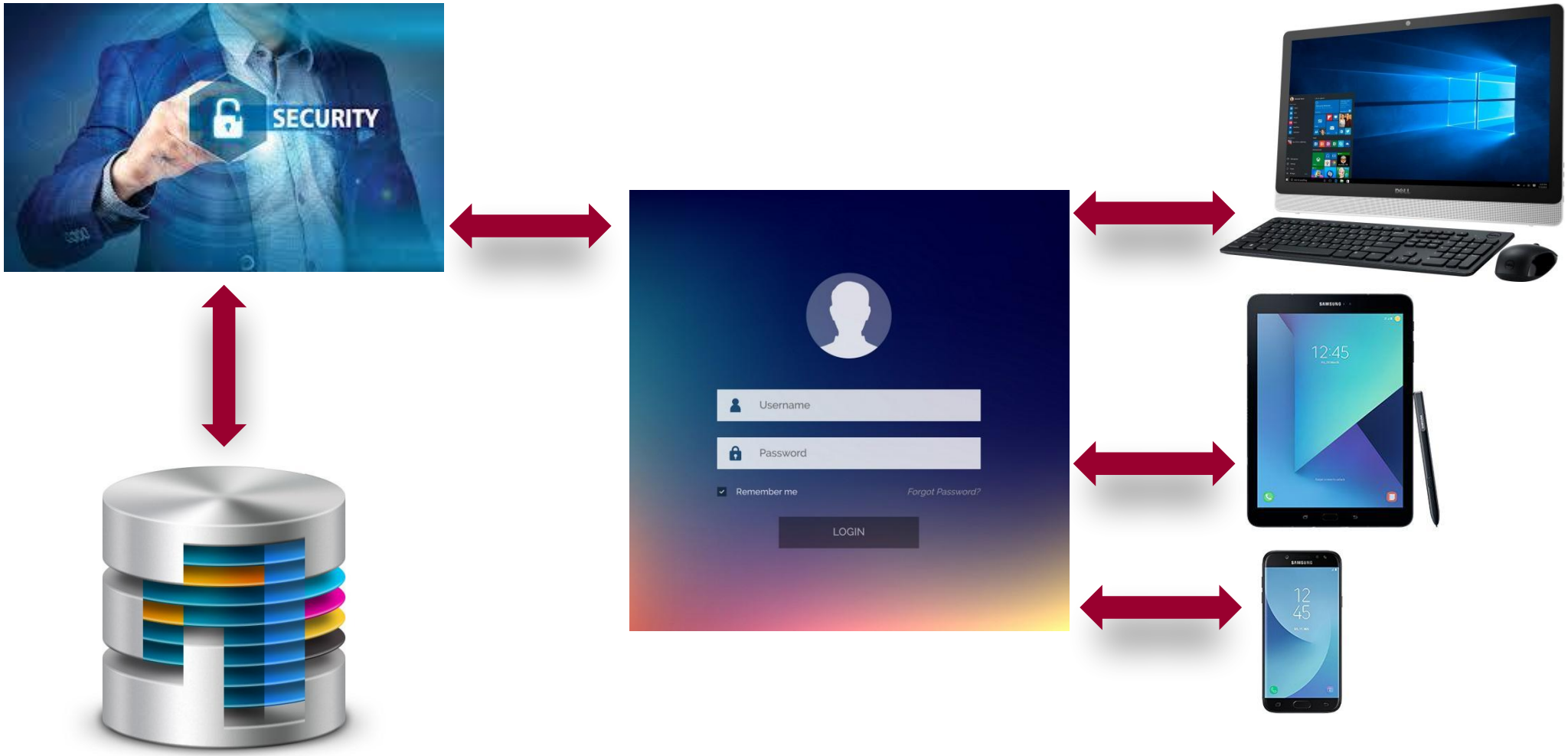


Lo sprint, di solito è di due settimane (10 gg lavorativi).

Al termine dello Sprint deve essere consegnato un Potentially Shippable Product Increment.

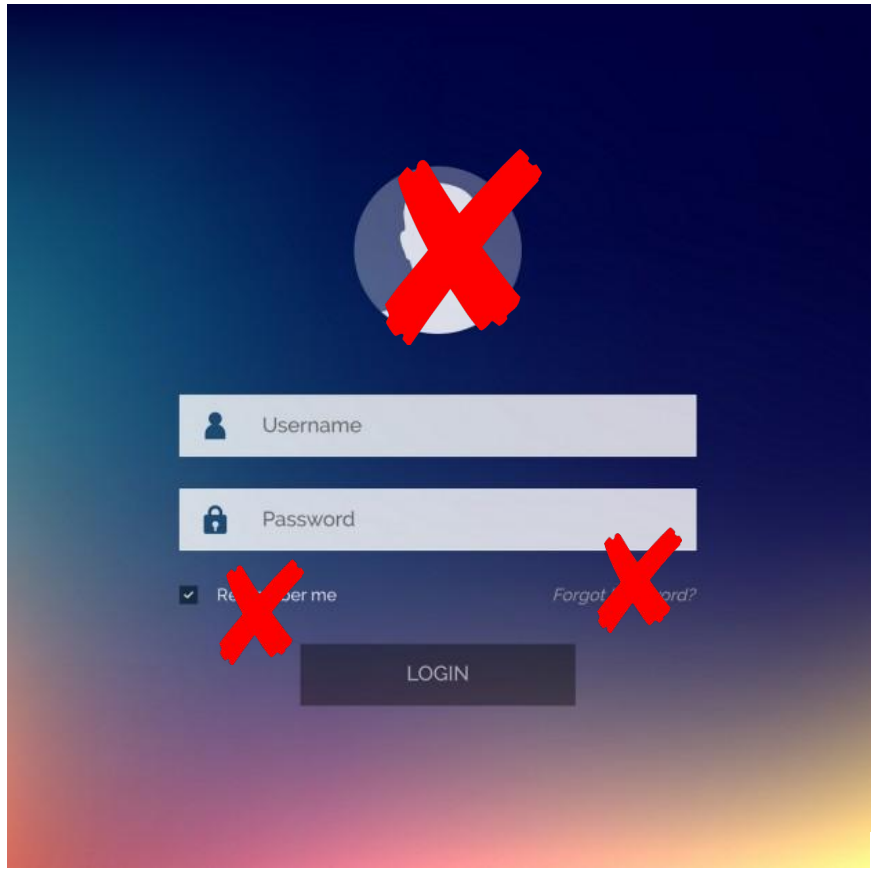
Vengono usati degli Information Radiators (es. Burndown/up chart, Sprint Kanban Board) per visualizzare la situazione al Team.

Potentially shippable product increment



Potentially shippable product increment

SE USERNAME = 12345
ALLORA «UTENTE ERRATO»
ALTRIMENTI
SE PASSWORD NON UGUALE «PIPPO»
ALLORA «PASSWORD ERRATA»



Daily Meeting

Ogni giorno il Team si riunisce per circa 15 minuti per allinearsi sugli sviluppi precedenti e concordare quelli successivi.

Nel Daily Meeting i Team Member comunicano allo Scrum Master gli Impediments che non sono riusciti a risolvere e lui li prende in carico.

Sprint Review

Al termine dello sprint il Team restituisce al Product Owner il lavoro fatto e lui lo accetta.

Alla Sprint Review può partecipare chiunque ed è un momento privilegiato per raccogliere feedback che possono essere usati per modificare il backlog prima dello Sprint Planning del giorno dopo

Sprint Retrospective

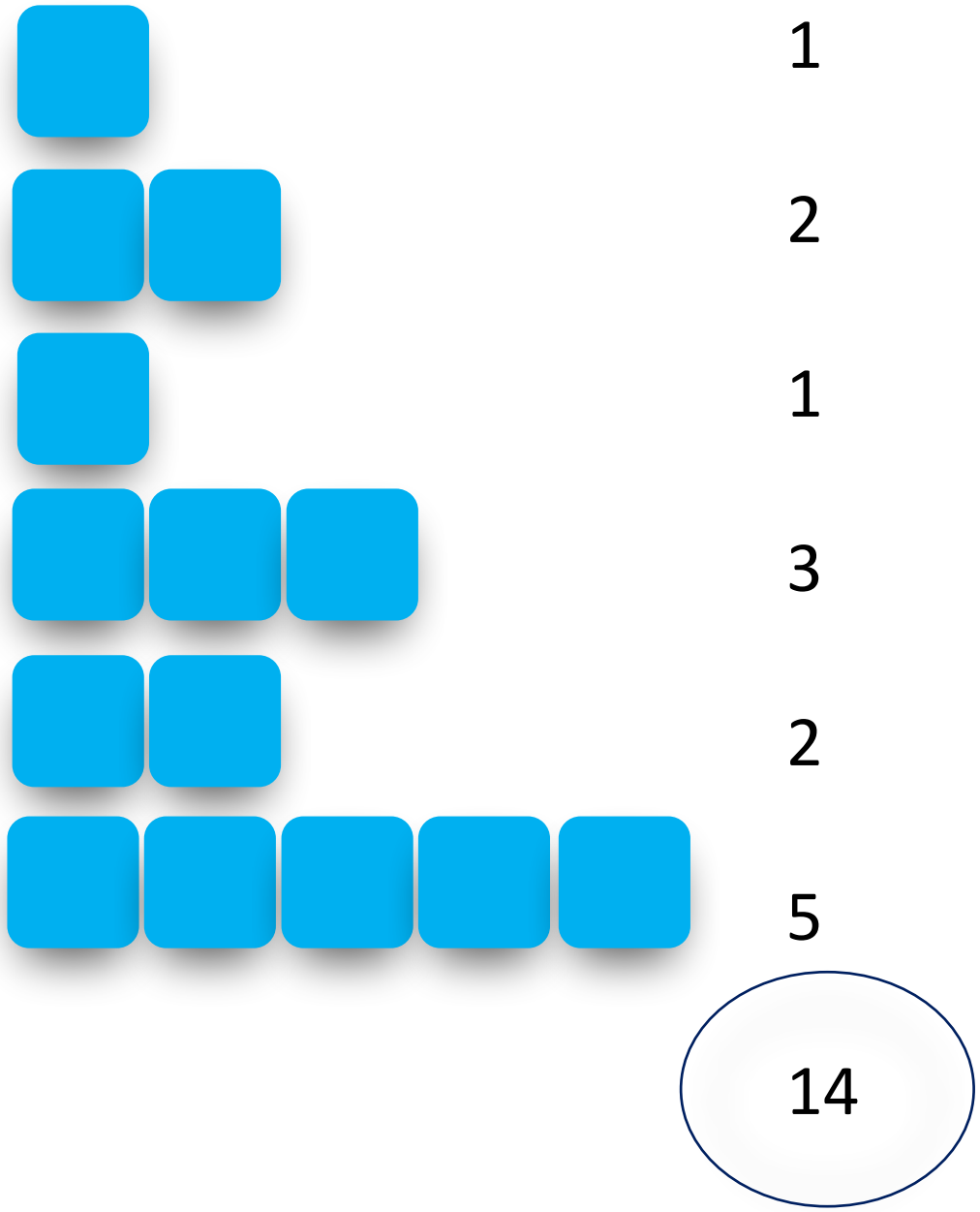
E’ il momento del Kay Zen.

Partecipa tutto il Team (PO, SM e Member).

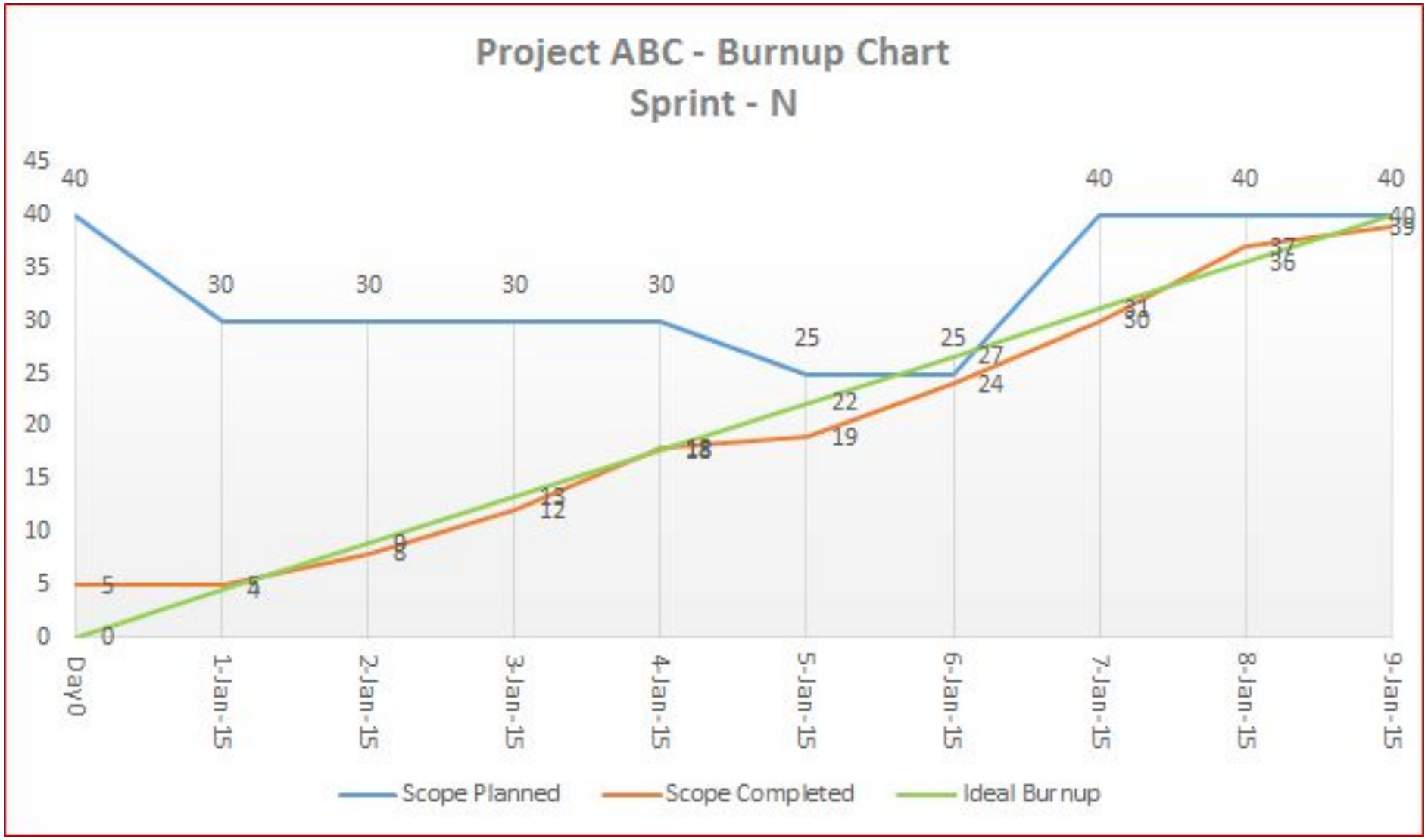
Si riguarda lo Sprint appena finito, da un punto di vista metodologico, e si individuano azioni di miglioramento che verranno messe in atto già dallo Sprint successivo.

L’analisi riguarda il COME e non il COSA.

Ma quando finiamo?



Pianificazione di progetto



Come è fatta una storia

Io _____

In qualità di _____

Desidero _____

Al fine di _____

Come è fatta una storia

Io Davide Roitero

In qualità di cliente registrato della Coop di Pisa

Desidero avere una applicazione dove compilare la mia lista della spesa ritrovandola poi sul mio dispositivo Salvatempo

Al fine di non dimenticare nulla e fare velocemente i miei acquisti

Criteri di accettazione

- I miei familiari possono proporre le cose che desiderano e io devo approvarle
- Il mio salvatempo mi deve proporre la lista della spesa ordinata per corsie espositive
- Il mio salvatempo deve ricordarmi le cose che non ho acquistato
- Durante la spesa vorrei che mi proponesse i prodotti in offerta alternativi a quelli da me scelti

La storia deve essere I.N.V.E.S.T.

Definition of Ready

I.N.V.E.S.T. +

- User Story defined
- User Story Acceptance Criteria defined
- User Story dependencies identified
- User Story sized by Delivery Team
- Scrum Team accepts UE artefacts
- Performance criteria identified, where appropriate
- Person who will accept the User Story is identified
- Team has a good idea what it will mean to Demo the User Story

La D.o.R. è elaborata dal Team con il P.O. e identifica le caratteristiche che deve avere una storia per essere accettata in uno Sprint Planning.

Quando il Team matura i suoi criteri diventano via via più restrittivi.

Definition of Done & P.S.P.I.

POTENTIALLY SHIPPABLE

+ TEST
+ STATIC ANALYSIS
+ INTEGRATION
+ PACKAGING
+ STAGING
+ CUSTOMER DOCUMENTATION
+ RISK EVALUATION
+ CODE
+ APPROVAL
+ UPDATE MARKETING MATERIAL
+ REGULATION
+ PREPARE FOR CUSTOMER FEEDBACK

UNIT
INTEGRATION
USER ACCEPTANCE
ACCEPTANCE
SYSTEM
PERFORMANCE
STABILITY
USABILITY
STRESS
MONKEY
SMOKE

UNDERLINED ONES ARE THE DEFINITION OF DONE

La D.o.D. è elaborata dal Team con il P.O. e identifica le caratteristiche che deve avere una storia per essere accettata in uno Sprint Review.

Quando il Team matura i suoi criteri diventano via via più restrittivi.