

AGILE – Per chi vuole saperne di più



Agile & Metriche del Software



CALENDARIO

14 GENNAIO 2014 13.00-14.00	1° Webinar "Agile Project Management" Contraria sunt complementa		
11 FEBBRAIO 2014 13.00-14.00	2° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (1a parte)		
11 MARZO 2014 13.00-14.00	3° Webinar "Agile Project Management" Agile è il futuro? Quando applicare una metodologia innovativa		
8 APRILE 2014 13.00-14.00	4° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (2a parte)		
20 MAGGIO 2014 13.00-14.00	5° Webinar "Agile Project Management" Il Tempo è denaro: Benefici economici dell'Agile		
10 GIUGNO 2014 13.00-14.00	6° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (3a Parte)		
8 LUGLIO 2014 13.00-14.00	7° Webinar "Agile Project Management" La metodologia Scrum		
		16 SETTEMBRE 2014 13.00-14.00	8° Webinar "Agile Project Management" Agile & Metriche del Software
		14 OTTOBRE 2014 13.00-14.00	9° Webinar "Agile Project Management" Come cambia l'organizza-zione e l'azienda
		11 NOVEMBRE 2014 13.00-14.00	10° Webinar "Agile Project Management" Il cambio del paradigma. Come cambiare il contesto intorno a noi
		9 DICEMBRE 2014 13.00-14.00	11° Webinar "Agile Project Management" Un caso di successo. Intervista a chi l'agile l'ha usato

- **1 PDU (registration on <http://pmi-rome.org>)**
- **Linkedin group: PMI Rome Italy Chapter**



Alessandro Puzielli

email: alessandro.puzielli@alepuzio.net

mobile: +39.338.61.95.249

LinkedIn public profile: <http://it.linkedin.com/in/alessandropuzielli/>

Profile Summary

- IT Consultant / J2EE Analyst Programmer
- Membro GUFPI-ISMA, IEEE, IEEE Computer Society, GdL "IT System Administrator" in AICA
- Ingegneria Informatica al Politecnico di Milano
- <http://www.alepuzio.net>



CALENDARIO

14 GENNAIO 2014 13.00-14.00	1° Webinar "Agile Project Management" Contraria sunt complementa
11 FEBBRAIO 2014 13.00-14.00	2° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (1a parte)
11 MARZO 2014 13.00-14.00	3° Webinar "Agile Project Management" Agile è il futuro? Quando applicare una metodologia innovativa
8 APRILE 2014 13.00-14.00	4° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (2a parte)
20 MAGGIO 2014 13.00-14.00	5° Webinar "Agile Project Management" Il Tempo è denaro: Benefici economici dell'Agile
10 GIUGNO 2014 13.00-14.00	6° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (3a Parte)
8 LUGLIO 2014 13.00-14.00	7° Webinar "Agile Project Management" La metodologia Scrum

- **1 PDU (registration on <http://pmi-rome.org>)**
- **Linkedin group: PMI Rome Italy Chapter**

16 SETTEMBRE 2014 13.00-14.00	8° Webinar "Agile Project Management" Agile & Metriche del Software
14 OTTOBRE 2014 13.00-14.00	9° Webinar "Agile Project Management" Come cambia l'organizza-zione e l'azienda
11 NOVEMBRE 2014 13.00-14.00	10° Webinar "Agile Project Management" Il cambio del paradigma. Come cambiare il contesto intorno a noi
9 DICEMBRE 2014 13.00-14.00	11° Webinar "Agile Project Management" Un caso di successo. Intervista a chi l'agile l'ha usato



Per trovare interventi sulla misurazione, all'interno nel mondo Agile, bisogna scavare un po'.

In genere le varie metodologie Agile sono ridotte scritte non formali di test e scusa ufficiale per evitare la redazione di documentazione estensiva.

In realtà, misurazione e metodologia Agile sono decisamente collegate, anche se le misurazioni tipicamente usate in ambiente Agile si discostano da quelle classiche (Function Points, COSMIC, etc).

L'approccio Agile si basa su osservazioni empiriche e su processi rapidi/non complessi che permettono uno scarto minimo tra stima e valori effettivi.

Mancano metriche in senso assoluto: sono considerate solo metriche relative.

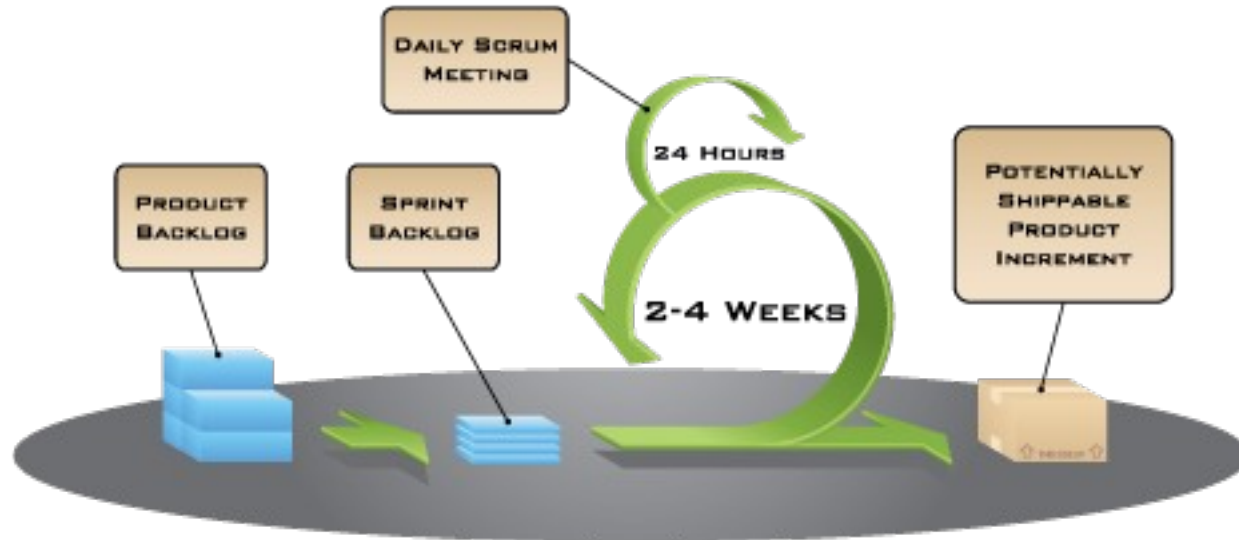


- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche



- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche
- Question time
- Recap & Lesson learned





COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

User Story & Story Points
Planning Poker
Burn Down Chart
Sprint interference
Kanban
WIP



User Story è la descrizione della singola funzionalità da implementare senza dettagli tecnici (tranne quelli con impatto funzionale), dal punto di vista dell'utente.

- 1. Come** spettatore di un webinar (*tipologia dell'utente*)
- 2. Devo poter** scaricare il file audio (*dominio non ambiguo*)
- 3. Così da** poterlo riascoltare su lettore mp3 (*funzionalità a valore aggiunto richiesta*)
- 4. Priority: 2** (*importanza della storia per l'utente finale*)
- 5. Story Points: 13** (*effort stimato per rendere utilizzabile la User Story*)



Henry Ford

Perché senza dettagli tecnici?

"Se avessi chiesto alle persone che cosa desideravano, mi avrebbero risposto che gli servivano solo cavalli più veloci".

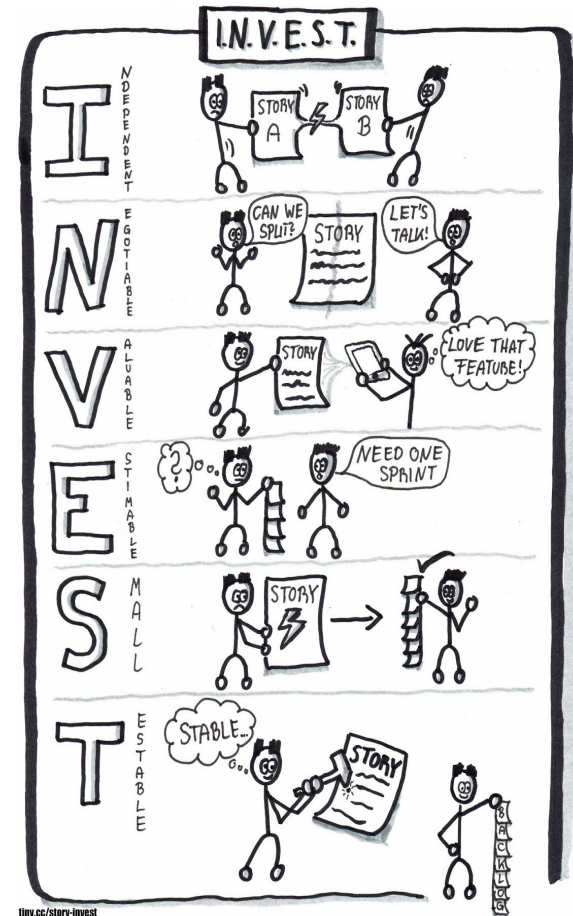
Quanto deve essere grande una User Story senza essere enorme? => Quanto deve essere grande un requisito?



La User Story utilizzabile deve essere

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

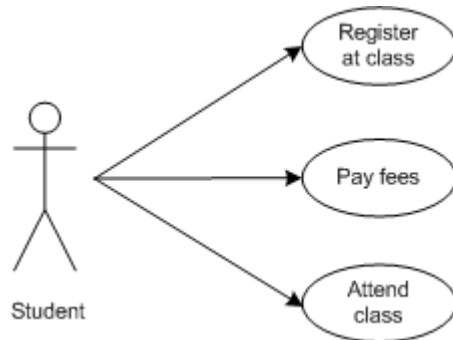
1. *Independent* (preferibilmente non dipendente da o accoppiata con altre storie)
2. *Negotiable* (negoziabile in termini di dettaglio tecnico e comportamento)
3. *Valuable* (dare valore aggiunto all'utilizzatore)
4. *Estimable* (essere stimabile dal team)
5. *Sized-properly / Small* (di dimensioni ridotte per essere sviluppata con altro nell'iterazione)
6. *Testable* (essere testabile)



E' un diagramma UML che rappresenta il comportamento di un sistema (4+1 views)

Similitudini

- Descrizioni concise.
- Nessun riferimento alla tecnologia (salvo casi particolari).
- Comprensibili intuitivamente senza una formazione onerosa.



As **who** I want
what so that
why

Differenze

1. Gli Use Case mostrano relazioni tra attori e tra altri Use Case, le Use Story sono atomiche (I di INVEST).
2. Gli Use Case non mostrano la parte funzionale del sistema, le User Story sì.
3. Gli Use Case hanno una sintassi formale, le User Story un template customizzabile.



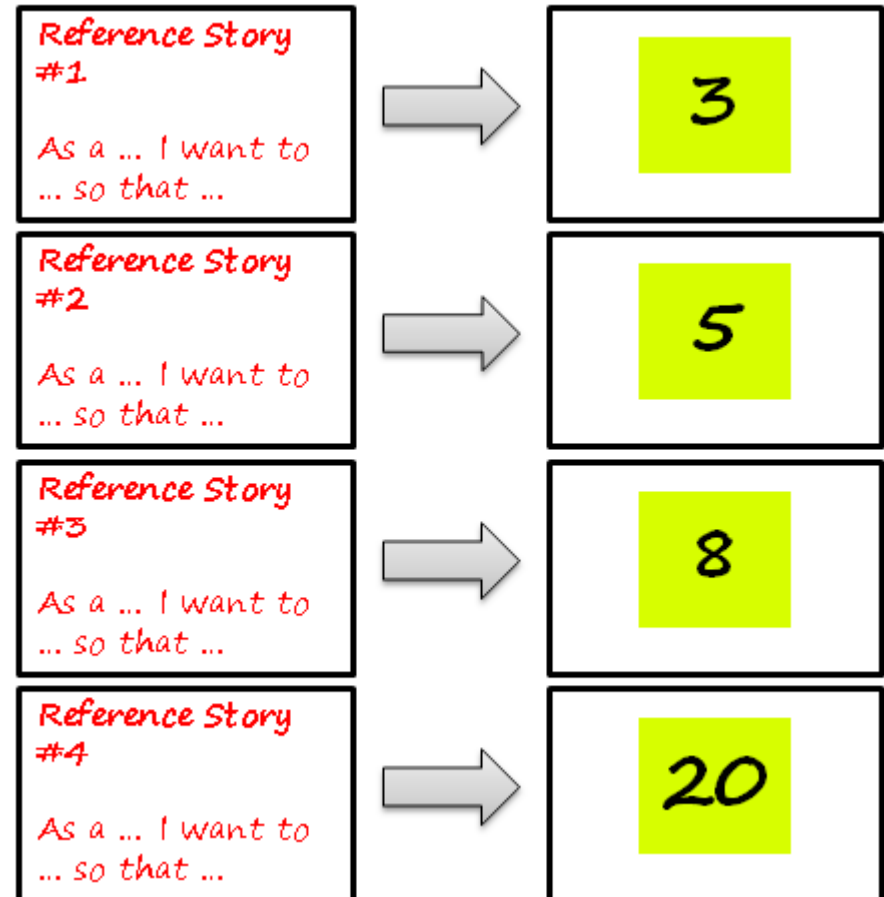
E' un metodo di conteggio gli Story Points di una User Story.

Interessa tutte le specializzazioni del team (sviluppatori, funzionali, tester, sistemisti, DBA, etc)

Ciascuno assegnerà un numero alla User Story: è una misura relativa e non assoluta. Durante le iterazioni si tenderà a convergere.

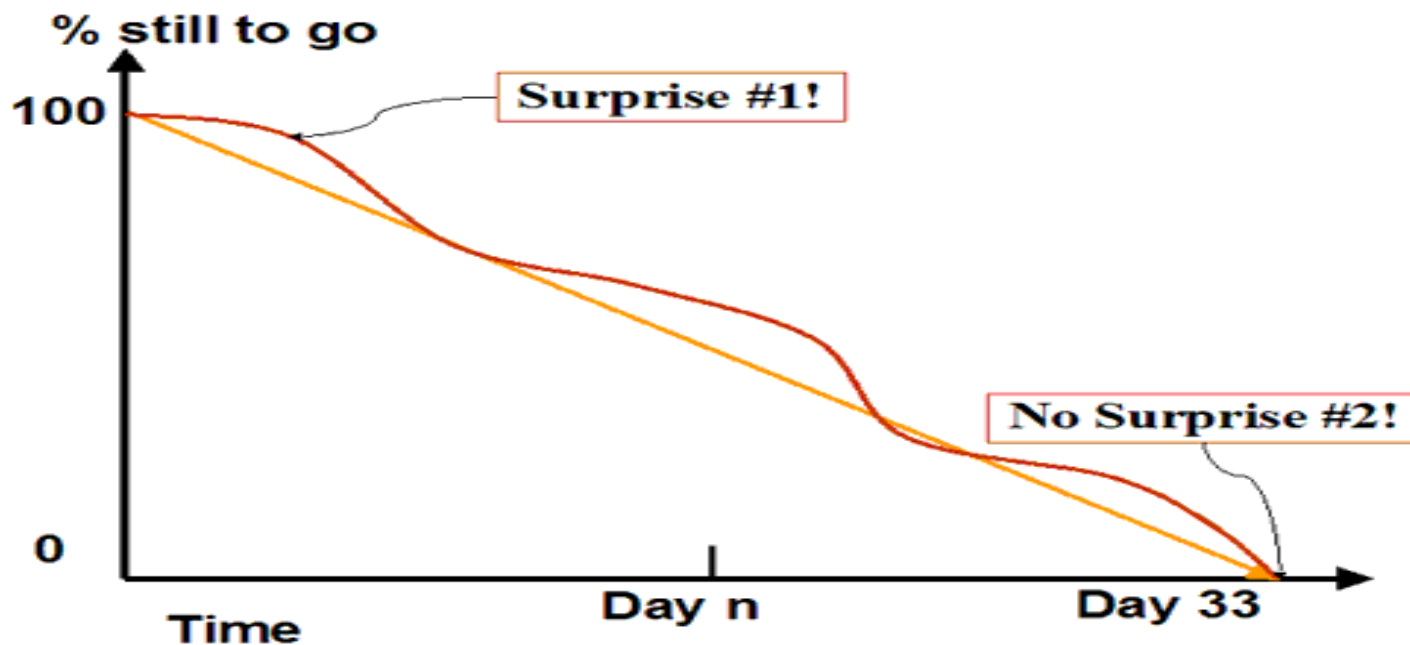
Visione cross-funzionale di tutti gli attori coinvolti: la discussione permette di far emergere i problemi nel processo di consegna.

La durata delle singole iterazioni non è modificabile.



Cosa serve?

Rappresenta gli scostamenti effettivi dell'andamento del progetto da quanto ipotizzato inizialmente



Cosa ci dice?

Ci mostra come evolve il progetto sotto una prospettiva puramente temporale, segnalando i picchi di lavoro.



Come crearlo?

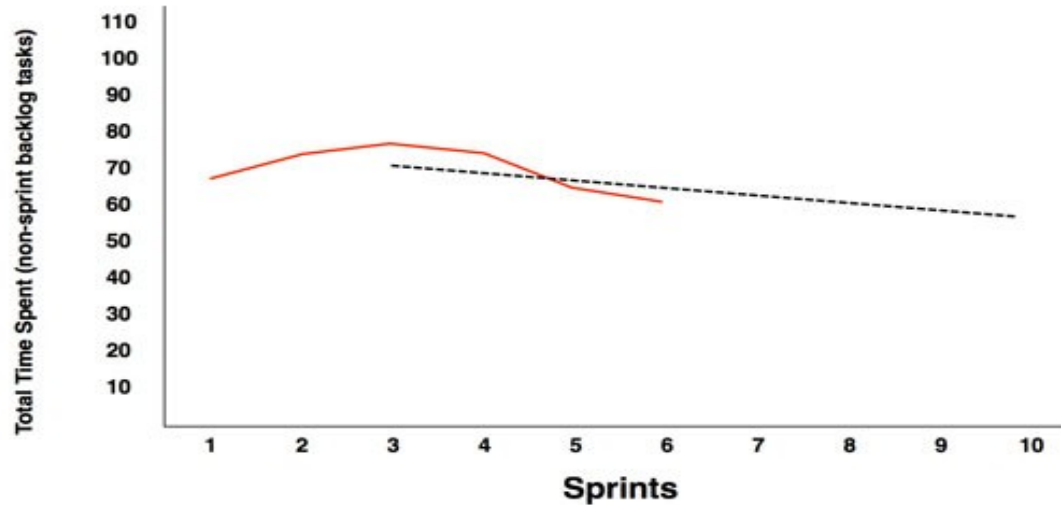
- 1) Per ogni giorno dello sprint, si disegna il totale dei tempi rimanenti di tutti i task
- 2) Si disegna una linea di interpolazione tra il totale del giorno corrente e quello del giorno prima
- 3) Si confronta con la linea prevista di *effort* che parte dal tempo 0 (*effort* max) fino alla data di consegna (*effort* 0)

Come leggerlo?

A ogni rilascio (iterazione) si confronta l'andamento effettivo dei Story Points/ore consegnate con l'andamento previsto inizialmente:

- 1) Se tale punto è sotto la linea ideale allora c'è un rallentamento per troppo lavoro, è il caso di ridurre il carico di lavoro.
- 2) Se tale punto è sopra la linea ideale allora è possibile, considerando il contesto, inserire nuove feature da sviluppare.





Cosa serve?

E' una metrica di produttività per aiutare i team a definire la pianificazione della propria capacità di sprint visualizzando e quantificando ciò che la impedisce.

Cosa ci dice?

Visualizzando come tempo perso da interruzioni dello sprint, questa metrica ci aiuta a precisare la capacità potenziale del prossimo sprint. Utile nel caso di pianificazione di sprint *commitment-based* (non tutti gli Sprint non sono definiti all'inizio del progetto)



Come crearlo?

- A. Per ciascun punto, disegna la somma del tempo speso da ogni componente del team per ogni task svolto non appartenente a quello sprint.
- B. Disegna una linea di andamento che interpola i punti.

Come leggerlo?

Come un Burn Down normale, solo che un'ordinata elevata indica problemi e non un elevato numero di Story Points.



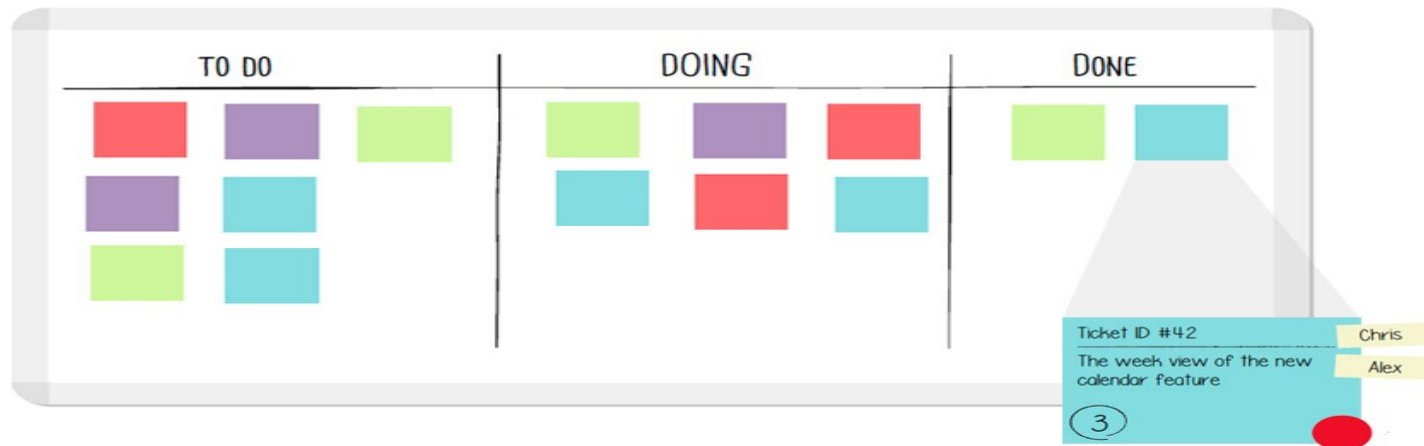
Cosa serve?

Serve a visualizzare, con uno sguardo, la situazione del flusso di lavoro

Come crearlo?

Divisione in più colonne (stati di lavorazione): ogni colonna ha un numero fisso di task in svolgimento (**WIP**: Work In Progress).

Se in uno stato di lavorazione il numero di task è superiore al WIP, allora bisogna bilanciare il flusso di lavoro



Il WIP non può essere calcolato *a-priori*, ma è da aggiornare iterativamente.



Cycle time / production lead time: Tempo trascorso dal momento in cui un team inizia a lavorare su un'attività (“Doing”) e fino al momento in cui tale attività è consegnata (“Done”): in pratica si misura quanto reattivo è il team nello sviluppo.

Customer lead time: Tempo trascorso dall'invio della richiesta del cliente nel backlog (“TO DO”) fino al momento in cui la richiesta soddisfatta è consegnata (“Done”). Questa metrica indica il grado di reattività dell'intero processo in base alla necessità del cliente.

Throughput: Tipicamente è il numero di storie/caratteristiche consegnate in una iterazione. Alcuni team le misurano pesate in base a qualche criterio (es. Function Points, LOC) in modo da rendere l'analisi più granulare

Work in Progress (WIP): Numero di task attualmente in corso nell'intero ciclo di sviluppo. Da sola non è molto utile, ma lo è se associata ad altre. Esempio: **Legge di Little (Average Cycle Time = WIP / Average Throughput)**






Numero empirico massimo di task in fase di lavorazione nell'intero processo.

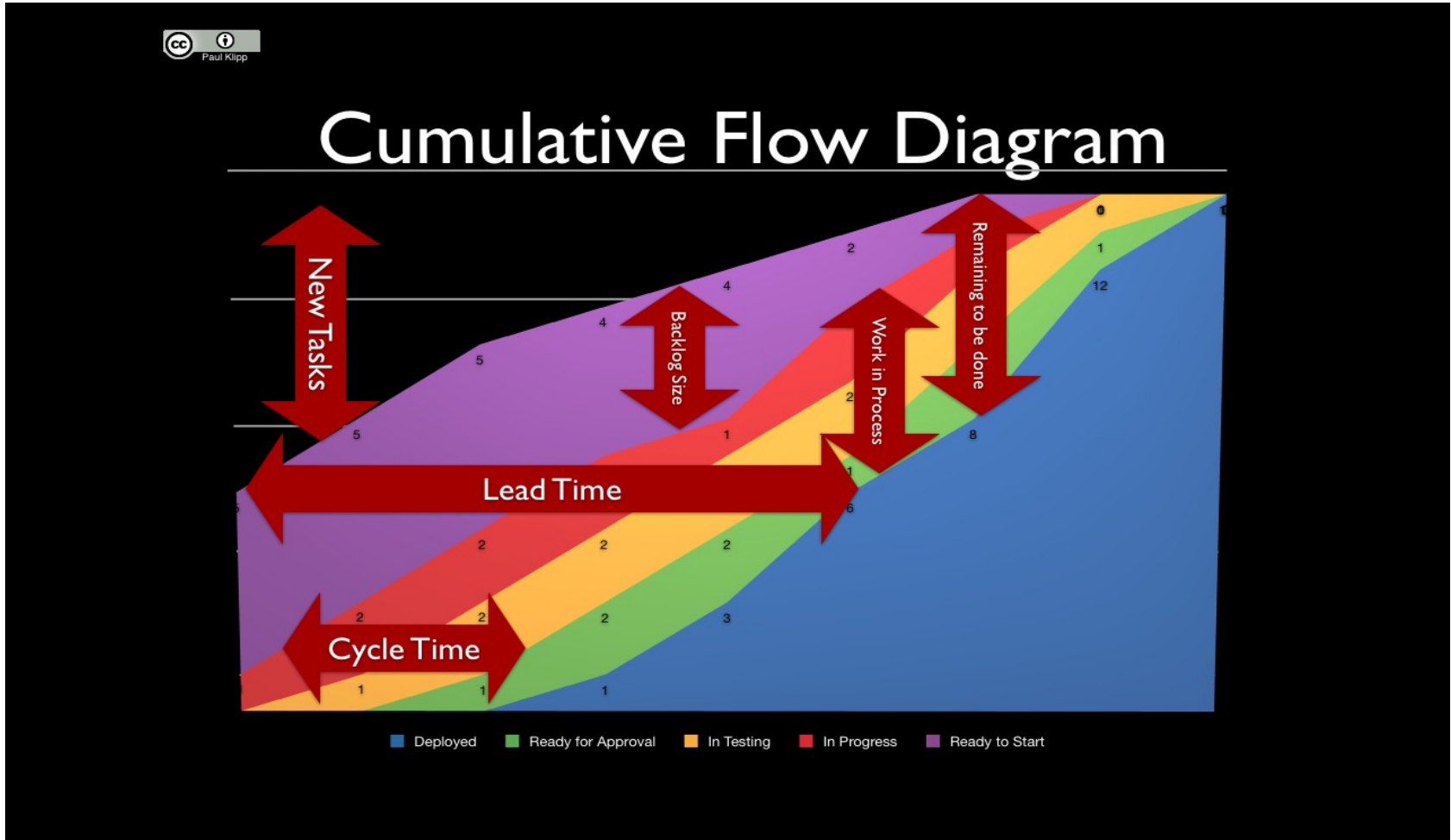
I task possono appartenere a più fasi e in genere si

Il principio cardine è *stop starting, start finishing*

Purtroppo può essere verificato solo empiricamente in base al team e all'organizzazione interna: in genere si stabilizza con 3 iterazioni

Backlog	Selected (5)	Develop (5)		Integrate & Test (3)		Release (3)	Retrospective (3)	Live
		In Progress	Done	In Progress	Done			
<div style="display: flex; flex-wrap: wrap; gap: 5px;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;">K</div> <div style="border: 1px solid black; padding: 2px; margin: 2px;">M</div> <div style="border: 1px solid black; padding: 2px; margin: 2px;">L</div> <div style="border: 1px solid black; padding: 2px; margin: 2px;">N</div> </div>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">J</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">H</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">G</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">I</div> </div>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">F</div>   </div>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">D</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">E</div> </div>		<div style="border: 1px solid black; padding: 5px; margin: 5px; width: fit-content;"> <p>Let's do a retro. We have hit the WIP</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px; width: fit-content;"> <p>Was glad the way development went. No real problems.</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px; width: fit-content;"> <p>Yeah - but we found a number of problems with B that shouldn't have been there.</p> </div>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">A</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">B</div> <div style="border: 1px solid black; padding: 2px; width: 30px; height: 20px; margin: 2px;">C</div> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px; width: fit-content;"> <p>Lets write more tests. From now on we will collaborate on writing acceptance tests before coding</p> </div>		
EXPEDITE (1)								





Cosa serve?

Mostra una fotografia dell'andamento del progetto da cui derivare alcune misurazioni classiche.

Cosa ci dice?

Come evolve storicamente il flusso del lavoro nel progetto, attraverso le varie fasi.

Come crearlo?

- 1) Ad ogni iterazione, segnare gli Story Points/ore lavorati in ogni stato del ciclo di lavorazione
- 2) Interpolare i nuovi punti con quelli precedenti e appartenenti al medesimo stato (“New”, “Doing”)

Come leggerlo?

Lungo l'asse verticale è possibile verificare quanti task sono in esecuzione (WIP), individuando possibili colli di bottiglia in corrispondenza di allargamenti dell'area.

Lungo l'asse orizzontale è possibile verificare il Cycle Time, ossia il tempo di permanenza di un task dal Backlog al rilascio.



La riduzione di spessore di banda - riducendo WIP (Work in Progress) o tempo di attesa - ha effetti visibili.

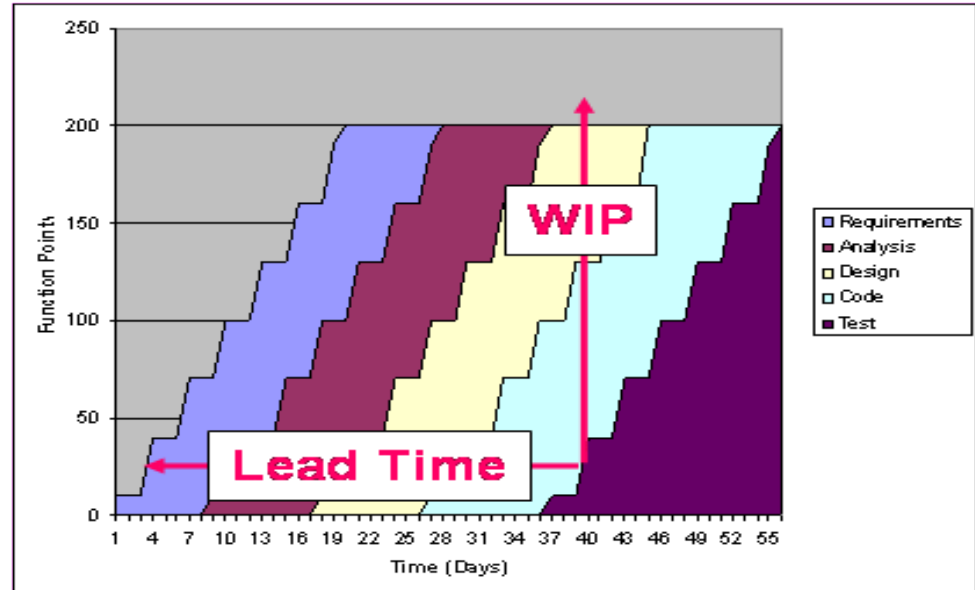
L'aumento dello spessore di qualunque banda deriva da un collo di bottiglia nel flusso che si concretizzerà nell'iterazione successiva.

Questo collo di bottiglia, visibile a tutto il team, provocherà una discussione KAIZEN (di miglioramento) all'interno del gruppo in modo da correggere il flusso di lavoro.

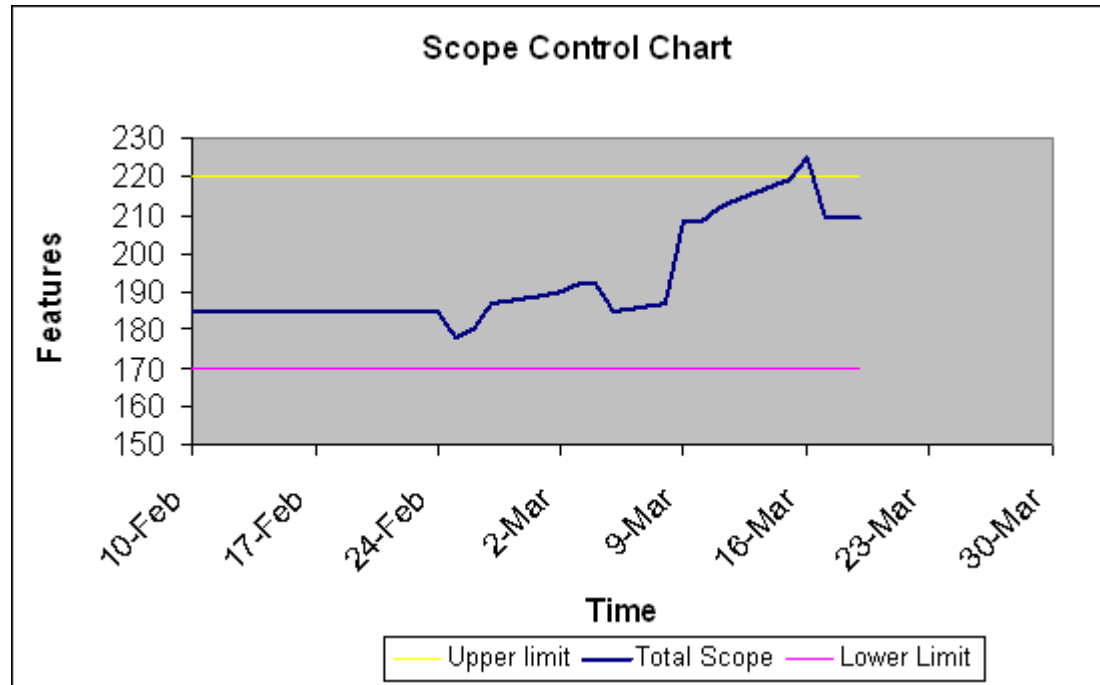
Lo WIP, misurato lungo l'asse verticale, impatta direttamente i valori del Cycle time e Lead time. Riducendo lo WIP assieme al bilanciamento delle operazioni si limitano i picchi del flusso di lavoro

E' possibile estrarre varie metriche *ad-hoc*

- **|started-designed|** == incertezza dei requisiti
- **|designed-scored|** == refactoring
- L'aumento dell'area del *Coding* indica un collo di bottiglia da risolvere (il lavoro è meno fluido)
- La distanza tra “Not started” e “Delivered” è il Lead Time, cioè il tempo per attraversare il flusso



Serve a verificare se la quantità di feature assegnata è in linea o meno con la velocità degli Sprint.



Le Control chart aiutano a tenere sotto controllo l'andamento del progetto, considerando i buffer. La linea centrale (in genere la media dei campioni disegnati) indica l'andamento teorico ottimo, gli Upper e lower bound segnano i limiti delle cause delle variazioni comuni, di solito un 3-sigma. Quando un processo è sotto controllo, i dati sono distribuiti e il 99.7% cadono all'interno dell'area 3-sigma. Quando i dati cadono fuori, allora ci sono impedimenti con cause particolari/eccezionali che rallentano fortemente l'efficienza del processo.



Burnout time: somma dei tempi relativi ai task consegnati + somma dei tempi dei problemi momentanei – tempi stimati precedentemente

Average velocity: $\sum (\text{velocità degli Sprint}) / (\#\text{Sprint})$

Estimated Time of Delivery Project: $\sum_{(\text{Backlog})} (\text{Story Points}) / (\text{Average Velocity})$

Velocity of Team: $\sum (\text{Story Points dell'iterazione attuale})$ o, anche, $\sum (\text{Story Points consegnati nella iterazione precedente})$

Estimated Velocity: velocità totale prevista alla prima iterazione

Actual Velocity: velocità totale aggiornata a ogni iterazione

Impediment Impact: $\sum_{(\text{ticket})} (\text{tempo perso}) * (\#\text{ore perse dai membri del team, in media})$

Task Estimation Accuracy Metric: $(\text{Estimated Velocity} / \text{Actual Velocity}) + \text{Remaining Story Points}$

Delivery Lead Time: quanto tempo spende il singolo task in una singola stazione (“TODO”, “Doing”, etc)



- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche
- Question time
- Recap & Lesson learned



- 1) Sviluppo Iterativo
- 2) TDD
- 3) Programmazione a coppie
- 4) Continuous integration



© Scott Adams, Inc./Dist. by UFS, Inc.



Focalizzate su metriche di prodotto (non di processo) automatizzabili: la stima serve per decidere, non per monitorare un progetto (*Martin Fowler*)

Bugs/LOC: quanto codice è necessario per avere un bug?

Code coverage: quanto codice sorgente è effettivamente usato in produzione?

Cyclomatic complexity: quanto è complesso l'algoritmo?

Program execution time: quanto è rapido l'applicativo a svolgere le attività funzionali?

Instruction path length: quante sono le istruzioni (originariamente codice macchina) per eseguire il task da parte dell'utente?

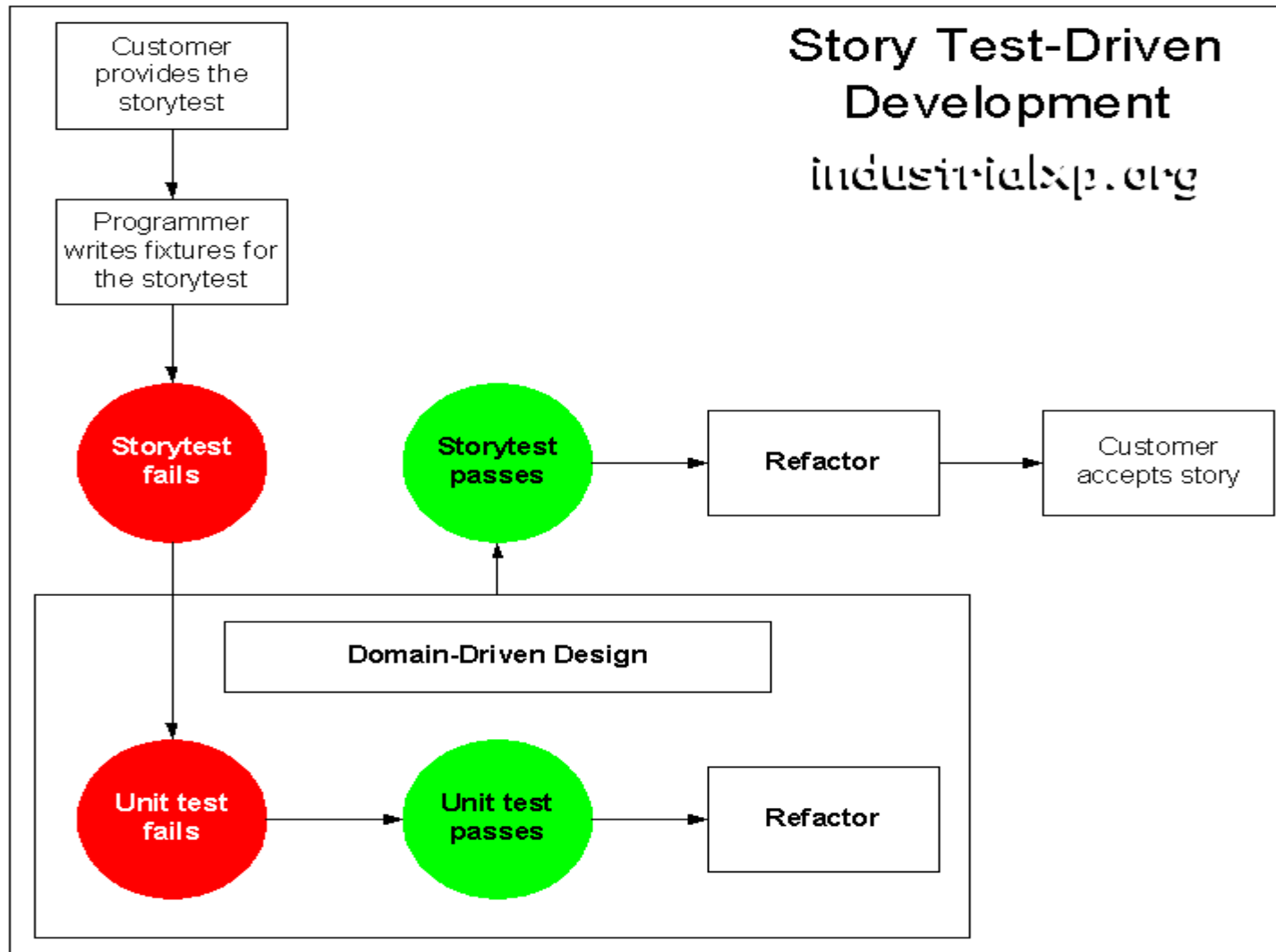
Comment density: CLOC/LOC: quanti commenti espliciti ci sono nel codice? 100% non è una buona situazione

XP Velocity: quanto lavoro è stato fatto nel progetto finora? E' la somma delle stime delle User Storie terminate.

Filosofia “Yesterdays Weather”

Le misurazioni devono richiedere processi semplici perchè forniscono approssimazioni che si basano sugli step precedenti





Lo sviluppo è guidato dai test definiti (TDD) o dal comportamento desiderato dalla prospettiva funzionale (BDD)

TDD: dapprima si dichiarano i test che rappresentano il comportamento del sistema, dal punto di vista implementativo.

BDD: dapprima si definiscono componenti software che modellizzano il comportamento del sistema, dal punto di vista funzionale.

Misure associate

Essenzialmente metriche di prodotto, non di processo (quindi, nessun Kanban, Story Points, etc)

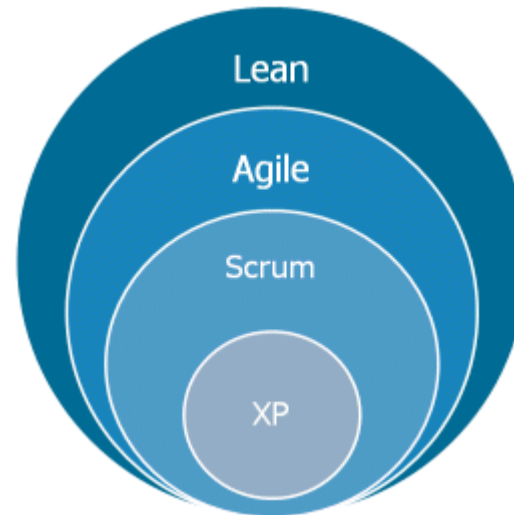
Il numero di test verificati in un qualunque momento indica il grado di avanzamento (Quanto è grande un test? Come un requisito? ==> *Quanto deve essere grande un requisito?*)

XP non ha un sistema di misurazione intrinseco a 360°, bisogna appoggiarsi ad altro e adattarlo.



- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche
- Question time
- Recap & Lesson learned





Il manifesto Agile è stato pubblicato nel 2001.

I framework derivati tendono o a una certa strutturazione organizzativa (es. SCRUM) o a una reattività ai *change request* (es. XP).

Da qualche anno la gestione informatica si confronta con i concetti gestionali dell'industria metalmeccanica e, tra i maggiori “prestiti”, vi è la metodologia Lean della Toyota, convertita in Lean Software Development.



Cycle Time (C/T) = tempo per completare una singola unità di operazione

First Time Through (FTT) = % di lavori completati correttamente la prima volta che sono processati

Demand = numero medio di unità per turno

Batch Size (BS) = dimensione di un lotto tipico quando viene considerato come singola unità

Takt time = frequenza di rilascio della richiesta del cliente = quantità di richiesta del cliente / periodo temporale

Throughput time = totale dei ritardi e del tempo del processo.

Process Ratio = tempo totale del processo / Throughput time

Value Added Ratio = Total VAT / Throughput time



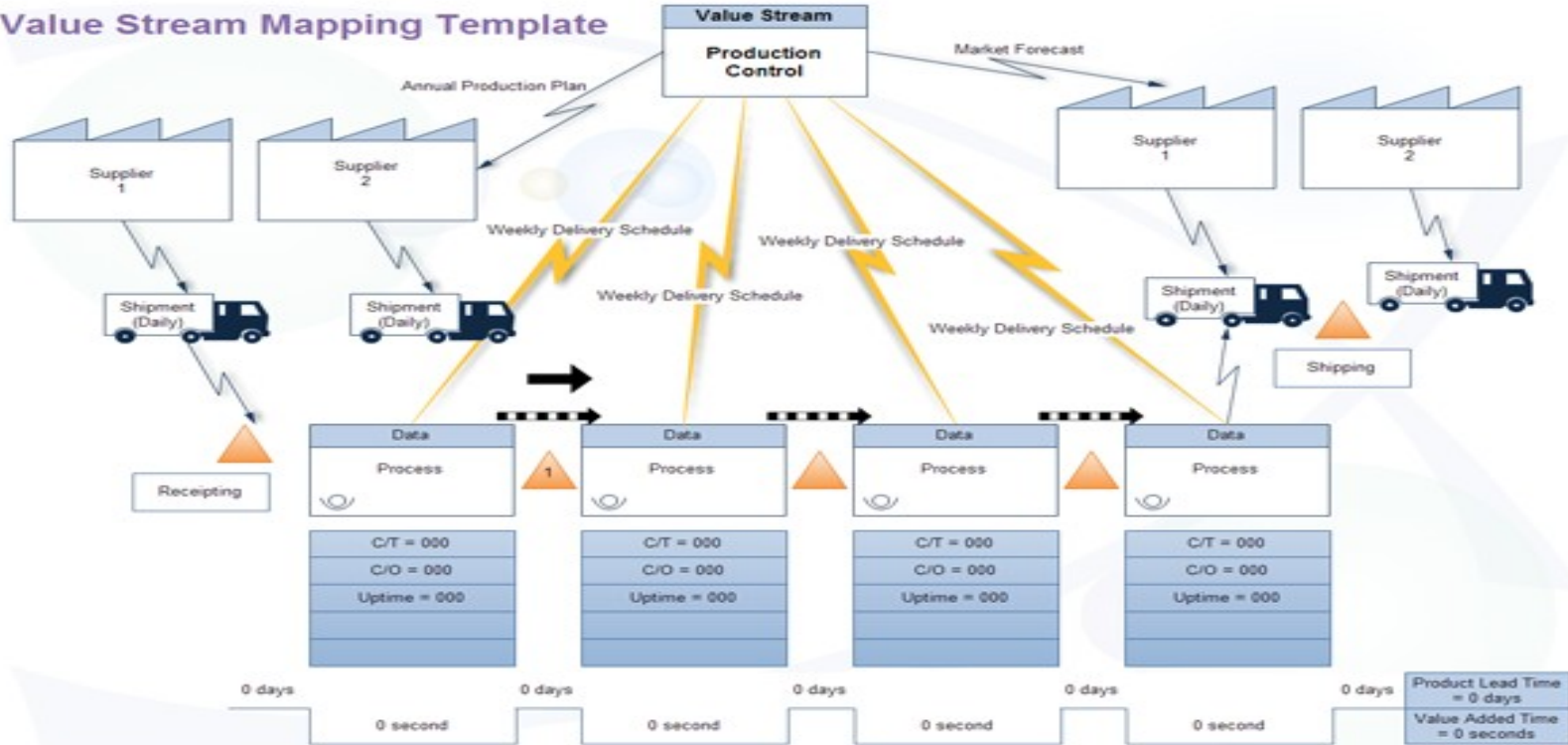


Serve e a confrontare i tempi di lavorazione di singole fasi del flusso di produzione, in modo da individuare anomalie rispetto il takt-time.

Sull'asse verticale è indicato il tempo, sull'orizzontale le varie operazioni: le fasi da modificare sono caratterizzate un tempo di processo superiore al takt-time (tempo di creazione del prodotto in base alle necessità del cliente)



Value Stream Mapping Template



Tempo a valore aggiunto (cycle time)

Lead time totale = tempo per completare una singola unità di operazione



Considerando l'efficienza del flusso come rapporto tra lavoro a valore aggiunto lead time (VA/LT), in genere si ottiene 1%, nei casi migliori 5%.

Per verificare l'andamento dell'intero flusso, e un sistema di calcolo economico che tenga ne conto, mi baserò su un esempio adattato dal libro “Progettare il valore” del blog Encob (<http://www.encob.net>)

Con tale sistema l'impatto del tempo e di colli di bottiglia è maggiormente visibile.



TEMPO [S]	Contatto con il cliente	Analisi	Testing	Implementazione	Rilascio e integrazione altri sistemi	Verifica in produzione	Totale
NUMERO TICKET	1	4	4	4	2	1	16
TEMPO [H] PER TICKET	3	15	12	25	30	36	121
CAPACITA [TICKET/H]	0,333333333	0,266666667	0,333333333	0,16	0,066666667	0,027777778	2,016666667
ORE LAVORATE A SETTIMANA (1/2 PAUSA)	37,5						
TICKET/SETTIMANA	12,5	10	12,5	6	2,5	1,041666667	44,54166667
EFFICIENZA	0,86	0,92	0,85	0,98	0,75	0,8	NA
TICKET EFFETTIVI/SETTIMANA	10,75	9,2	10,625	5,88	1,875	0,833333333	39,16333333
COSTO PRODUTTIVO SETT	4000						
TICKET IN ATTESA	0	5	0	2	2	2	11
TICKET PROCESSATI	1	1	1	1	1	1	6
TICKET TOTALI	1	6	1	3	3	3	17
COSTO UNITARIO DI UN SINGOLO TICKET	372,0930233	434,7826087	434,7826087	680,2721088	2133,3333333	4800	8855,263683
LEAD TIME[H]	3	22,5	3	18,75	45	108	8,429752066
SETTIMANE DI PRODUZIONE PRESENTI IN MOMENTO T DEL CICLO	0,2247933884						
SPESA OPERATIVA SETTIMANALE	4000						
SPESA OPERATIVA IMPEGNATA IN UN MOMENTO T DEL FLUSSO	899,173553719						
FLOW COSTING	202.1282699269						
Legenda	DATI CALCOLATI	DATI A PRIORI	DATI TOTALI				



Dati di partenza

- Numero di richieste (ticket) degli utenti in un anno = 630
- Numero di settimane lavorative durante l'anno = 45 (315 giorni)
- Numero di ticket al giorno = $630/315 = 2$ ticket/giorno
- Ore lavorate a settimana (pausa pranzo di 30 minuti) = $5 \text{ g} * 7,5 \text{ h/g} = 37,5 \text{ h}$
- Ore disponibili in un anno = $37,5\text{h/sett} * 45 \text{ sett/anno} = 1687,5 \text{ h}$
- Forza lavoro: team da 4 ingegneri del software
- Spesa operativa settimanale = $4\text{cad.} * 200 \text{ euro/g/cad.} * 5 \text{ gg} = 4000 \text{ euro}$



Calcolo classico – no Flow Costing

- *Inventario (costo della manodopera) annuale* = 800 euro * 45 sett = 36000 euro/anno
- *Costo singolo ticket* = Quantità inventario / Domanda giornaliera cliente = 36000 euro/anno / 630 ticket/anno = 57,14 euro/ticket
- *Costo di attività di un giorno* = Costo singolo ticket * numero persone * num. Ticket giornalieri = 57,14 euro/ticket* 2ticket/giorno = 114,28 euro



Formule varie del Flow Costing

Capacità lavorativa oraria = 1 ora/(tempo/singolo ticket)

Numero pezzi lavorati a settimana

•Teorici (efficienza = 100%) = cap lav oraria * num ore/settimana = cap. lav. Ora. * 40

•Effettivi (efficienza < 100%) = cap lav oraria * num ore/settimana * efficienza (%)

Takt time = numero richieste annuali/ore settimana = (630 ticket/1687 h) = 0,37 **ticket/h**

Numero di batch presenti in un dato momento del ciclo di sviluppo = Somma dei batch in attesa e processati (qua si vede l'accumulo)

Lead Time = somma totale dei cycle time / ore settimanali

Capitale impegnato in un dato momento della produzione = (spesa operativa settimanale) *
(#sett di produzione in un dato momento)

Flow costing = importo capitale impegnato nel flusso di produzione / takt time

Adesso l'allocazione dei costi dipende anche dalla quantità di batch (Funzionalità? Story Points? Function Points? Decide il contesto => “*Quanto è grande un requisito?*”) e da quanto velocemente essi sono lavorati.

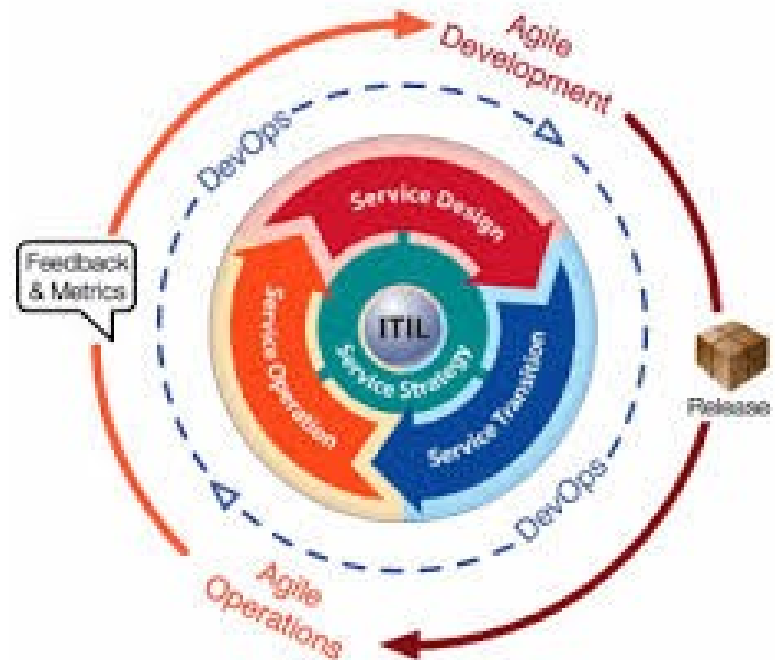
Il calcolo base invece lavora solo sulle risorse disponibili e non varia col variare dei processi.



Integrazione delle attività tra sviluppatori e sistemisti e/o personale delle operazioni IT.

Metriche su:

- Numero e frequenza di rilasci software
- Volume di difetti
- Tempistiche/costi per release
- Numero e frequenza delle indisponibilità / cali di performance
- Impatti sui guadagni per le indisponibilità / cali di performance
- Numero e costi delle risorse
- Ricezione dei requisiti e loro gestione
- Sviluppo Agile
- Creazione software eseguibile
- Rilasci e distribuzione
- Test di unità
- Test di accettazione
- Quality Assurance
- Monitoraggio delle performance
- Cloud



- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche
- Question time
- Recap & Lesson learned





- Metriche tipiche in SCRUM
- Metriche tipiche in XP
- Cenni sul mondo Lean Software Development e sulle relative metriche
- Question time
- Recap & Lesson learned



- I processi Agile sono processi distribuiti e rivolti alla prototipazione
- Le misurazioni sono usate più per monitorare che per progettare: fare il contrario in *short-time* comporterà ricicli del processo di sviluppo all'aumento delle feature.
- Differenti set di metriche per differenti obiettivi
- **SCRUM**
 - Metriche dedicate sulla puntualità nell'iterazione time-boxed
- **XP**
 - Metriche concentrate sulla qualità del prodotto
- **LEAN SOFTWARE DEVELOPMENT**
 - Metriche concentrate sul continuo improvement e riduzione sprechi locali
- **Quali tipi di metriche possono essere considerate?**
 - Metriche dedicate sui processi, non sulle persone (es. produttività)
 - Metriche facilmente automatizzabili a livello software
 - Metriche basate sul contesto (contratto, SLA, conoscenza pregressa, etc)

Prossime lezioni

Come cambia l'organizzazione a seguito del cambio di concetto di qualità del progetto?



Metriche SCRUM

- <http://scaledagileframework.com/metrics>
- <http://agileatals.org/articles/item/metrics-for-a-scrum-team>

Metriche XP

- <http://martinfowler.com/tags/metrics.html>

Metriche LEAN

- <http://www.slideshare.net/yyeret/lean-agile-metrics-and-kpis>
- Encob.net:
 - Il Flow Costing originale è contenuto nel libro “Progettare il valore”



Thank you!



CALENDARIO

14 GENNAIO 2014 13.00-14.00	1° Webinar "Agile Project Management" Contraria sunt complementa		
11 FEBBRAIO 2014 13.00-14.00	2° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (1a parte)		
11 MARZO 2014 13.00-14.00	3° Webinar "Agile Project Management" Agile è il futuro? Quando applicare una metodologia innovativa		
8 APRILE 2014 13.00-14.00	4° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (2a parte)		
20 MAGGIO 2014 13.00-14.00	5° Webinar "Agile Project Management" Il Tempo è denaro: Benefici economici dell'Agile		
10 GIUGNO 2014 13.00-14.00	6° Webinar "Agile Project Management" Agile & Requirement Management: Quanto è grande un requisito? (3a Parte)		
8 LUGLIO 2014 13.00-14.00	7° Webinar "Agile Project Management" La metodologia Scrum		
		16 SETTEMBRE 2014 13.00-14.00	8° Webinar "Agile Project Management" Agile & Metriche del Software
		14 OTTOBRE 2014 13.00-14.00	9° Webinar "Agile Project Management" Come cambia l'organizza-zione e l'azienda
		11 NOVEMBRE 2014 13.00-14.00	10° Webinar "Agile Project Management" Il cambio del paradigma. Come cambiare il contesto intorno a noi
		9 DICEMBRE 2014 13.00-14.00	11° Webinar "Agile Project Management" Un caso di successo. Intervista a chi l'agile l'ha usato

- **1 PDU (registration on <http://pmi-rome.org>)**
- **Linkedin group: PMI Rome Italy Chapter**

