

SCRUM

611

Copyright © Ing. Alessandro DI GIOVANNI

Cosa è SCRUM



- ▶ Nota come **metodologia agile** per lo sviluppo del software
- ▶ E' un **framework**
- ▶ Il suo motto è: "**l'arte del possibile**"
- ▶ Prevede di dividere il progetto in blocchi rapidi di lavoro (**Sprint**) alla fine di ciascuno dei quali creare un **incremento del software**
- ▶ È un metodo di gestione dei prodotti (non dei progetti)
- ▶ Oggi distribuito da *Advanced Development Methods* (<http://www.controlchaos.com/>)

612

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: origine del termine



- ▶ SCRUM NON è un acronimo...
- ▶ ...bensì la parola SCRUM è presa dal corrispondente termine del Rugby
 - ▶ indica il **pacchetto di mischia ordinata** (gruppo di persone che si spinge per agguantare la palla)
 - ▶ è evidente metafora del team di sviluppo che deve lavorare insieme in modo che tutti gli attori del progetto spingano nella stessa direzione, agendo come un'unica entità coordinata.



613

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: alcune informazioni

- ▶ Lo si scrive solitamente in lettere maiuscole (perchè così faceva Schwaber nei suoi primi articoli)
- ▶ SCRUM si basa sull'**empirismo**
L'empirismo afferma che:
 1. la conoscenza deriva dall'esperienza
 2. le decisioni si basano su ciò che si conosce
- ▶ Per implementarlo serve **processo di trasformazione**
 1. della mentalità
 2. del modo di agire
- ▶ È "*facile da comprendere ma difficile da padroneggiare*"

614

Copyright © Ing. Alessandro DI GIOVANNI



SCRUM: le origini

- ▶ Ideato e sviluppato nel 1993 da *Jeff Sutherland*...
- ▶ ... che nel 1995 collaborò con *Ken Schwaber* per formalizzarla all'OOPSLA'95...
 - ▶ = Object-Oriented Programming, Systems, Languages & Applications
 - ▶ Conferenza annuale americana di ricerca in ambito OOP
 - ▶ Prima edizione nel 1986
 - ▶ In Europa esiste la **ECOOP** (European Conference on OOP)
- ▶ ...in cui venne essenzialmente documentato ciò che Ken e Jeff avevano appreso negli anni precedenti applicando Scrum.
- ▶ Nel 2001, *Ken Schwaber* ha collaborato con *Mike Beedle* per descrivere il metodo in un libro chiamato **Agile Software Development with Scrum**.
- ▶ Nel 2004, *Ken Schwaber* ha pubblicato il libro **Agile Project Management with Scrum** pubblicato da Microsoft Press.

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: le vere origini...




Hirotaka Takeuchi Ikujiro Nonaka

- ▶ Il termine Scrum, in quanto applicato allo sviluppo di prodotti, è stato per la 1° volta utilizzato nel 1995 in 2 testi di *Ikujiro Nonaka* e *Hirotaka Takeuchi* (editi dalla Oxford University Press):
 1. "New Product Development Game" (Harvard Business Review 86116:137-146, 1986) in cui venne citato
 2. "The Knowledge Creating Company" in cui venne elaborato
- ▶ Nel 1986, *Hirotaka Takeuchi* e *Ikujiro Nonaka* descrissero un **nuovo approccio allo sviluppo di prodotti commerciali** che avrebbe aumentato la velocità e la flessibilità, basato su casi di studio presi dall'industria automobilistica e quella relativa alla realizzazione di fotocopiatrici e stampanti. Essi lo chiamarono **approccio olistico o rugby**, in quanto l'intero processo viene eseguito da un team interfunzionale su più fasi che si sovrappongono, dove la squadra «cerca di raggiungere l'obiettivo come unità, passando la palla avanti e indietro».

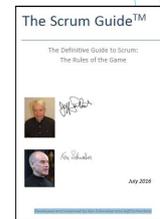
616

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: guide ufficiali

▶ The Scrum Guide

- ▶ Developed and sustained by *Ken Schwaber* and *Jeff Sutherland*
- ▶ 17 pagine
- ▶ Ultima edizione del luglio 2016



▶ SBOK Guide (A Guide to the Scrum Body of Knowledge)

- ▶ Disponibile gratuitamente in PDF (o a \$35,95 cartacea)
(<https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016.pdf>)
- ▶ 340 pagine
- ▶ Ultima edizione del 2016



617

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: Definizione ufficiale

«Scrum è un **framework di processo** utilizzato dai primi anni novanta per gestire lo sviluppo di prodotti complessi. Scrum non è un processo o una tecnica per costruire prodotti ma piuttosto è un framework all'interno del quale è possibile utilizzare vari processi e tecniche. Scrum rende chiara l'efficacia relativa del proprio product management e delle proprie pratiche di sviluppo così da poterle migliorare.»

(*Jeff Sutherland - La Guida a Scrum™*)

618

Copyright © Ing. Alessandro DI GIOVANNI

Diffusione di SCRUM

- ▶ E' tra i più diffusi metodi di APM
- ▶ Nel 2009 l'86% degli sviluppi Agile erano basati su SCRUM
- ▶ Nel 2017, dal *The 11th annual state of agile report*, risulta che:
 - ▶ il 94% delle aziende che hanno partecipato al survey dichiarano l'utilizzo di metodologie agili per la gestione dei progetti...
 - ▶ ...di queste, il 25% appartiene al comparto IT (Information Technology)
 - ▶ il 69% dichiara di utilizzare metodologie di project management agili per accelerare il processo di delivery...
 - ▶ ...il 58% usa SCRUM
 - ▶ il 98% dei progetti gestiti con metodologie agili termina con successo

619

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: è usato da...

- | | |
|------------------------------|-------------------------|
| ▶ Amazon | ▶ Microsoft |
| ▶ BBC | ▶ Nielsen Media |
| ▶ BMC Software | ▶ Nokia |
| ▶ Electronic Arts | ▶ Océ |
| ▶ Facebook | ▶ Philips |
| ▶ First American Real Estate | ▶ Sabre |
| ▶ Google | ▶ Salesforce.com |
| ▶ Intuit | ▶ Siemens |
| ▶ Ipswitch | ▶ Time Warner |
| ▶ John Deere | ▶ Turner Broadcasting |
| ▶ Lexis Nexis | ▶ Yahoo |
| ▶ Lockheed Martin | ▶ ... |
| ▶ MailUp | |

620

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: è usato per...

- ▶ Software commerciale
- ▶ Sviluppo in-house
- ▶ Sviluppi a contratto
- ▶ Progetti a prezzo prefissato
- ▶ Applicazioni finanza
- ▶ Applicazioni certificate ISO 9001
- ▶ Sistemi embedded
- ▶ the Joint Strike Fighter
- ▶ Applicazioni 24/7 con running time 99.999%
- ▶ Video giochi
- ▶ Sistemi life-critical
- ▶ Software di controllo satelliti
- ▶ Siti web
- ▶ Social Network (Facebook)
- ▶ E-Commerce (Amazon)
- ▶ Handheld software
- ▶ Sw per cellulari
- ▶ Network switching
- ▶ Applicazioni ISV
- ▶ Grandi sistemi
- ▶ ...

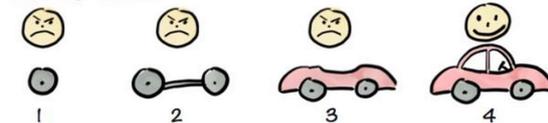
621

Copyright © Ing. Alessandro DI GIOVANNI

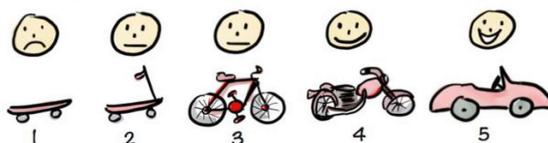
SCRUM: è usato dove...

- ▶ in contesti in cui è difficile pianificare in anticipo (la maggior parte dei progetti software!)
- ▶ ...e quindi si vuole fare questo:

Not like this....



Like this!



622

Henrik Kniberg

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: per diminuire i rischi che le cose possono cambiare...

▶ Iteratività

- ▶ ti organizzi il lavoro in scatole temporali di durata fissa, chiamate iterazioni.
- ▶ All'inizio di ciascuna iterazione fisserai degli obiettivi realizzativi parziali
- ▶ Al termine verificherai il risultato conseguito
- ▶ Acquisirai così man mano informazioni utili per pianificare il lavoro rimanente correggendo eventualmente il tiro

▶ Incrementalità

- ▶ non costruisci il tuo prodotto/servizio tutto insieme, a valle di una unica grande e dettagliata fase di analisi e progettazione, ma in maniera progressiva, un po' alla volta
- ▶ In questo modo puoi far fronte meglio ad esigenze che potrebbero cambiare nel tempo.

623

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM: Princìpi e Valori

SCRUM si basa su:

- ▶ **valori**
- ▶ **princìpi**

che saranno condivisi dal team di sviluppo

«Il lavoro in squadra insito in SCRUM contribuisce a rafforzare i valori, che non appaiono come imposti dall'alto, ma come princìpi condivisi e accettati mutualmente da tutto il personale coinvolto nello sviluppo»

624

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM

▶ I valori sono:

1. Concentrazione (Focus)
2. Coraggio (Courage)
3. Apertura (Openness)
4. Impegno (Commitment)
5. Rispetto (Respect)

(vedi dopo)

- ▶ L'assenza di valori porterebbe a un'implementazione meccanica, che porta poco lontano.

625

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM: Concentrazione (Focus)

- ▶ Per lavorare bene, bisogna concentrarsi solo su poche cose alla volta.
- ▶ Lavorare in squadra, concentrandosi su pochi elementi, migliora anche i tempi di consegna.

Because we focus on only a few things at a time,
we work well together and produce excellent work.
We deliver valuable items sooner.

626

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM: Coraggio (Courage)

- ▶ Lavorare in squadra, e non da soli, presuppone il sostegno dei compagni: avere il supporto e riconoscere negli altri membri delle risorse dà il coraggio per affrontare compiti impegnativi.

Because we work as a team,
we feel supported and
have more resources at our disposal.
This gives us the courage
to undertake greater challenges.

627

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM: Apertura (Openness)

- ▶ Lavorare in squadra presuppone che tutti si aprano agli altri membri del team, spiegando ciò che si sta facendo e gli eventuali problemi che si incontrano.
- ▶ Pur senza lamentarsi, è bene esprimere eventuali fondate preoccupazioni, in maniera tale che possano essere affrontate e risolte in gruppo.

As we work together,
we express how we're doing,
what's in our way,
and our concerns so they can be addressed.

628

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM: Impegno (Commitment)

- ▶ Ciascuno è artefice del suo destino:
per avere un controllo maggiore sulle nostre azioni e
per raggiungere gli scopi prefissimi,
è necessario un impegno, che è impegno verso la buona riuscita.

Because we have great control over our own destiny,
we are more committed to success.

629

Copyright © Ing. Alessandro DI GIOVANNI

I valori di SCRUM: Rispetto (Respect)

- ▶ Occorre rispettarsi vicendevolmente, e
aiutarsi tutti a essere sempre più “rispettabili”:
lavorando insieme, si “vince” e si perde “insieme” e
il rispetto è anche una conseguenza della presa di coscienza di questo fatto.

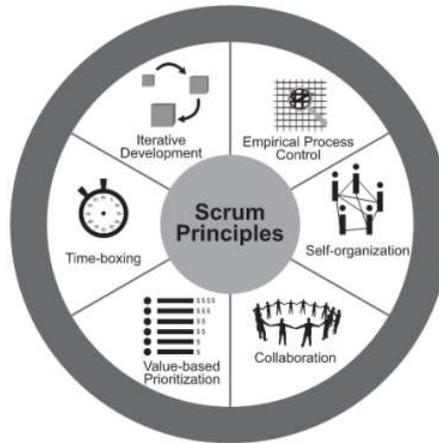
As we work together,
sharing successes and failures,
we come to respect each other and
to help each other become worthy of respect.

630

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. Empirical Process Control
2. Self-Organization
3. Collaboration
4. Value-based Prioritization
5. Time-boxing
6. Iterative Development



631

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. **Empirical Process Control**
 - ▶ le decisioni si prendono sulla base di osservazioni e sperimentazioni, anziché secondo una preventiva pianificazione di dettaglio
2. Self-Organization
3. Collaboration
4. Value-based Prioritization
5. Time-boxing
6. Iterative Development

632

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. Empirical Process Control
2. **Self-Organization**
 - ▶ SCRUM riconosce che i lavoratori di oggi hanno più conoscenza da offrire, anziché soltanto la loro esperienza tecnica, perciò rilasciano più valore se auto gestiti
3. Collaboration
4. Value-based Prioritization
5. Time-boxing
6. Iterative Development

633

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. Empirical Process Control
2. Self-Organization
3. **Collaboration**
 - ▶ Lo sviluppo del prodotto è un processo condiviso di creazione del valore d'opera di tutti gli stakeholder che lavorano e interagiscono per dare il massimo valore
4. Value-based Prioritization
5. Time-boxing
6. Iterative Development

634

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. Empirical Process Control
2. Self-Organization
3. Collaboration
4. **Value-based Prioritization**
 - ▶ Rilasciare il massimo valore nel più breve tempo possibile richiede prioritizzazione e scelta di ciò che va fatto e da dove va fatto
5. Time-boxing
6. Iterative Development

635

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

1. Empirical Process Control
2. Self-Organization
3. Collaboration
4. Value-based Prioritization
5. **Time-boxing**
 - ▶ Il tempo viene considerata una risorsa limitata per segnare il ritmo al quale tutti gli stakeholder lavorano e contribuiscono
6. Iterative Development

636

Copyright © Ing. Alessandro DI GIOVANNI

Principi di SCRUM

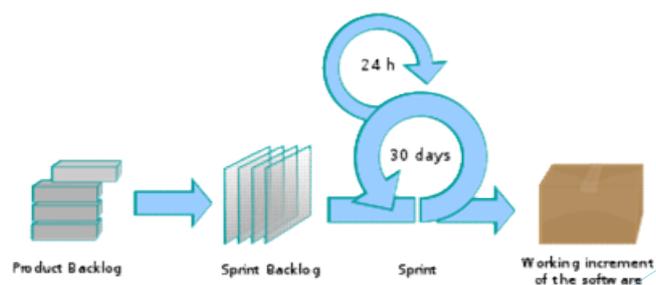
1. Empirical Process Control
2. Self-Organization
3. Collaboration
4. Value-based Prioritization
5. Time-boxing
6. **Iterative Development**
 - ▶ Il cliente non sempre è in grado di definire requisiti molto consistenti.
 - ▶ Il modello iterativo è più flessibile nell'accettare continue richieste di modifiche ai requisiti.

637

Copyright © Ing. Alessandro DI GIOVANNI

Contenuto del Framework

- ▶ È un **framework di processo** che contiene un insieme di:
 1. Roles (Ruoli)
 2. Events (Eventi) / Cerimonie
 3. Artifacts (Artefatti)



638

Copyright © Ing. Alessandro DI GIOVANNI

Vediamo SCRUM in meno di 15 minuti...



<https://www.youtube.com/watch?v=Wku93RUALqc>

...e poi entreremo nel dettaglio!

639

Copyright © Ing. Alessandro DI GIOVANNI

Roles (1)

In base alla natura del loro coinvolgimento nel processo di sviluppo, si dividono in 2 gruppi distinti:

1. **Suini («Pig»)** 
 - ▶ impegnati a costruire software regolarmente e frequentemente
2. **Polli** 
 - ▶ interessati al progetto, ma in realtà indifferenti, perché anche se fallissero non sono i maiali, cioè, quelli impegnati a realizzarlo

I bisogni, i desideri, le idee e le influenze dei ruoli di pollo vengono presi in considerazione, ma non gli consentito in alcun modo di incidere, falsare o governare il vero e proprio progetto Scrum

Nel 2011, si è scelto di non utilizzare più questa analogia, perché da molti è stata ritenuta inappropriata ed offensiva

640

Copyright © Ing. Alessandro DI GIOVANNI

Roles (2)

I 2 gruppi derivano il loro nome da una battuta su un maiale e un pollo riguardante l'apertura di un ristorante:



641

Copyright © Ing. Alessandro DI GIOVANNI

Ruoli core («maiali»)

SCRUM Team



1. **Scrum Master**
 - ▶ gestisce e garantisce il processo (tipicamente al posto del Project Manager)
 - N.B.: Non deve esserci un Project Manager!
2. **Product Owner**
 - ▶ Rappresenta i committenti e il business
 - ▶ Decide cosa fare
3. **Team**
 - ▶ gruppo inter-funzionale di 5-9 persone che fa tutto (analisi, progettazione, implementazione, test, ...)
 - ▶ Decide come fare

642

Copyright © Ing. Alessandro DI GIOVANNI

Ruoli non core («polli»)

1. Stakeholder

- ▶ persone che permettono il progetto e per i quali il progetto produce i benefici concordati che ne giustificano la produzione
- ▶ Sono coinvolti direttamente nel processo solo durante le Sprint Review (vedi dopo)

2. Fornitori

3. Manager / PMO (Scrum Guidance Body)

- ▶ Persone che controllano l'ambiente di lavoro



643

Copyright © Ing. Alessandro DI GIOVANNI

Roles: Product Owner

- ▶ Definisce la Vision del Progetto e la Schedulazione
- ▶ Rappresenta la «voce del cliente»
- ▶ Ha competenze di dominio
- ▶ Definisce i requisiti del cliente sotto forma di **Epiche** e **User Story**
 - ▶ Identifica le caratteristiche del prodotto e le traduce in lista di priorità (decide le priorità e le riassegna se necessario)
- ▶ Chiarisce i **requisiti** ai membri del team
- ▶ Stabilisce le **priorità** degli oggetti nel Backlog di Prodotto in base al valore di business.
- ▶ Decide i **rilasci**: date e contenuti
- ▶ Fornisce Acceptance Criteria (**DoD** - Definition of Done)
- ▶ Ha **responsabilità economica** (deve massimizzare il ROI)
- ▶ Accetta o rifiuta i risultati
- ▶ Può essere il cliente stesso
- ▶ Diverso dal Product Manager (perché è parte del Team e non delega solo lo sviluppo al Project Manager)



644

Copyright © Ing. Alessandro DI GIOVANNI

Roles: Development Team

- ▶ È **cross-funzionale**
(cioè include tutte le competenze necessarie a rilasciare il prodotto ad ogni sprint)
- ▶ È **autogestito / auto-organizzato**
 - ▶ scelgono come meglio compiere il lavoro organizzandosi e coordinandosi al proprio interno
- ▶ I membri sono tutti **componenti del team**
 - ▶ Non esistono ruoli come analista, DBA, ...
 - ▶ Tutti fanno tutto (a seconda delle loro specifiche abilità, ma imparando e mettendosi in gioco anche su punti in cui sono meno forti)
- ▶ Composto da 5 a 9 persone (solitamente 7)
- ▶ Sviluppa il prodotto e fornisce idee al Product Owner
- ▶ Dovrebbe essere allocato al 100% allo sprint corrente
- ▶ Sono responsabili del loro lavoro



645

Copyright © Ing. Alessandro DI GIOVANNI

Roles: Scrum Master

- ▶ È un «facilitatore», un «allenatore», un «insegnante»
- ▶ Aiuta il Dev Team
 - ▶ ad applicare SCRUM
 - ▶ a rimuovere gli ostacoli
- ▶ Assicura il benessere del Dev Team
 - ▶ Deve proteggere il Dev Team e tenerlo concentrato sui compiti (deve fare da cuscinetto per bloccare interferenze esterne)
- ▶ Non è manager del team, ma rappresenta il management
 - ▶ Non ha autorità sul team
 - ▶ Ha autorità sui processi
 - ▶ Non dice al team cosa fare
- ▶ Deve essere una persona dedicata full-time
- ▶ In team piccoli può essere un membro del team
- ▶ Non può essere anche Product Owner



646

Copyright © Ing. Alessandro DI GIOVANNI



Roles: Scrum Master - attributi

Secondo **Mike Cohn** un buon Scrum Master deve avere questi 6 attributi:

1. Responsabile
2. Umile
3. Collaborativo
4. Impegnato
5. Influyente
6. Competente

▶ Mike Cohn è uno dei fondatori della **Scrum Alliance** (<https://www.scrumalliance.org/>)



Copyright © Ing. Alessandro DI GIOVANNI

Artifacts

Artefatto = Opera (sia materiale che intellettuale) che deriva da un processo trasformativo intenzionale da parte dell'uomo.

1. Project Vision Statement
2. Sprint Goal
3. Impediment Log
4. Product Increment
5. Product Backlog
6. Sprint Backlog
7. Burndown chart

648

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Project Vision Statement

- ▶ Documento in cui vengono descritti:
 - ▶ il progetto
 - ▶ Il problema di business che il progetto intende risolvere
- ▶ Tipicamente fornito dal PO
- ▶ Viene stabilito all'avvio di un progetto focalizzando sul problema anziché sulla soluzione
- ▶ E' una visione idealistica del risultato atteso

649

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Sprint Goal

- ▶ Rappresenta l'obiettivo che il Team si impone di raggiungere con lo sprint
- ▶ Proposto dal PO e accettato dal Dev Team per ogni Sprint
- ▶ Non può cambiare nel corso dello Sprint

650

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Impediment Log

- ▶ Log di tutti gli impedimenti / ostacoli manifestati dai membri del team durante i Daily Meeting
- ▶ Registrazione a cura dello Scrum Master...
- ▶ ...che ha il compito di risolverli

651

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Product Increment

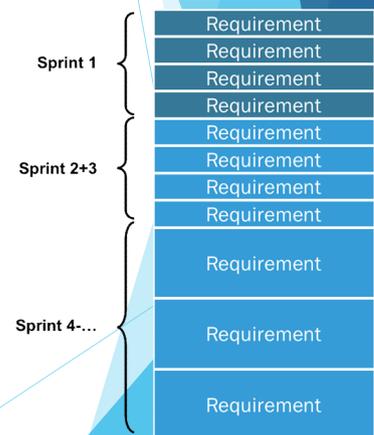
- ▶ È il risultato di uno sprint...
- ▶ ...potenzialmente consegnabile dal team
- ▶ ...che soddisfa i Criteri di Accettazione (done)

652

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Product Backlog

- ▶ E' la tabella di marcia del prodotto
- ▶ È la lista ordinata dei **requisiti** del prodotto (cosa deve essere fatto)
- ▶ Contiene tutte le attività da fare (To Do) per incrementare il valore del lavoro svolto
- ▶ Ciascun elemento è un Product Backlog Item (PBI)
- ▶ È ordinato per priorità (in alto priorità maggiore)
 - ▶ Definita dal PO
- ▶ Modificabile da tutti
- ▶ Il PO ne è responsabile
- ▶ Ciascun PBI è stimato dal Dev Team in **story point**
 - ▶ secondo la **successione di Fibonacci** arrotondata (0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?, ∞)
- ▶ Esiste e si evolve per tutta la durata del prodotto



653

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Product Backlog - DEEP

Un buon Product Backlog è DEEP

- ▶ **Detailed**
 - ▶ le voci a priorità maggiore saranno le più dettagliate
- ▶ **Estimated**
 - ▶ le voci devono essere stimate (dal Dev Team!) in termini di **story points** e vengono nuovamente stimate quando inserite nello Sprint Backlog
- ▶ **Emergent**
 - ▶ le voci possono essere continuamente riviste (aggiunte, eliminate, modificate come priorità)
- ▶ **Prioritized**
 - ▶ le voci hanno una priorità (che deve basarsi sul vantaggio economico e sul rischio)

654

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Product Backlog - Tipo di PBI

Il backlog può contenere diversi tipi di Product Backlog Item:

1. **Epic**
 - ▶ Macro-funzionalità
2. **User Story**
 - ▶ Funzionalità
3. **Task**
 - ▶ Dettaglio di una funzionalità
4. **Chore**
 - ▶ Attività tecnica priva di valore per il cliente
5. **Spike**
 - ▶ Attività di analisi priva di valore per il cliente
6. **Bug**
 - ▶ Errore o anomalia

655

Copyright © Ing. Alessandro DI GIOVANNI

User Story

- ▶ Corrisponde al Work Package del Project Mgmt «tradizionale»
- ▶ Ciascuna contiene:
 1. **Descrizione** in formato Who/What/Why
 - ▶ Definito come “As a user, I want” Template
 - ▶ **As a <type of user>, I want <some goal> so that <some reason>**
 - ▶ Es: *Come operatore voglio una stampa delle fatture per poterle inviare al cliente*
 2. **Acceptance Criteria**
 3. **Estimate**
- ▶ Ordinate in priorità decrescente nel Backlog (sia nello Sprint che nel Product)
- ▶ Priorità è data dal **Valore di business** (definito dal PO)

As a/an	I want to...	so that...
moderator	create a new game by entering a name and an optional description	I can start inviting estimators
moderator	invite estimators by giving them a url where they can access the game	we can start the game
estimator	Join a game by entering my name on the page I received the url for	I can participate
moderator	start a round by entering an item in a single multi-line text field	we can estimate it
estimator	see the item we're estimating	I know what I'm giving an estimate for
estimator	see all items we will try to estimate this session	I have a feel for the sizes of the various items
moderator	see all items we try to estimate this session	I can answer questions about the current story such as "does this include"
moderator	select an item to be estimated or re-estimated	the team sees that item and can estimate it

656

Copyright © Ing. Alessandro DI GIOVANNI

User Story - Modello INVEST

Una US deve soddisfare i seguenti requisiti:

- ▶ **Independent**
 - ▶ Non deve dipendere da nessuna altra storia
- ▶ **Negotiable**
 - ▶ Deve essere possibile chiarirla in tutti i suoi aspetti (ostacoli compresi)
- ▶ **Valuable**
 - ▶ Esiste solo se anche l'utente finale trae vantaggio
- ▶ **Estimable**
 - ▶ Deve essere possibile effettuarne una stima
- ▶ **Small**
 - ▶ Deve essere piccola
- ▶ **Testable**
 - ▶ Deve essere oggettivamente verificabile

657

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Product Backlog - iniziare a scriverlo

Per iniziare a scrivere il backlog:

- ▶ **Generate PERSONAS**
 - ▶ tempo da dedicare: 5 min (fare quello che si riesce in questo tempo)
- ▶ **Generate STORIES (EPICS)**
 - ▶ Tempo da dedicare: 15 min (fare quello che si riesce in questo tempo)
- ▶ **SPLIT STORIES (USER STORIES)**
 - ▶ Prendere massimo 2 epic e dividerle in 2 US ciascuna

658

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Sprint Backlog

- ▶ Lista di attività da svolgere in uno sprint
- ▶ Deciso dal Dev Team sulla base del backlog
- ▶ Le singole attività sono svolte in completa autonomia dal team
 - ▶ Il Lavoro non viene assegnato, ma richiesto su base «volontaria» da ciascun membro del Dev Team
- ▶ Solitamente tenuto aggiornato in una kanban board

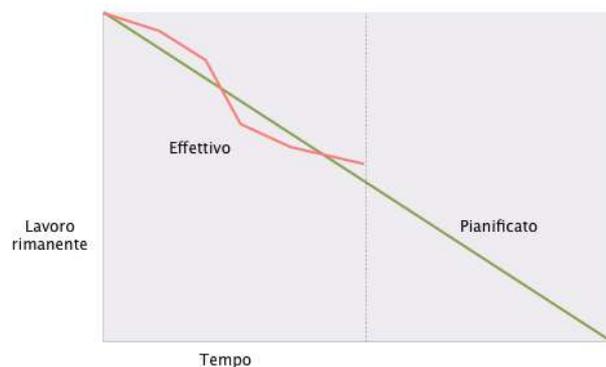
659

Copyright © Ing. Alessandro DI GIOVANNI

Artifacts: Burndown chart

Può essere:

- A. **Release Burndown Chart**
 - ▶ Aggiornato a fine di ogni Sprint
- B. **Sprint Burndown Chart**
 - ▶ Aggiornato ogni giorno



- ▶ È un grafico che rappresenta il lavoro che deve essere ancora completato
- ▶ Dà una semplice visione di come procedono i lavori
- ▶ Deve essere pubblico
- ▶ Lavoro rimanente può essere descritto con qualsiasi unità di misura (solitamente si usano gli story point)

660

Copyright © Ing. Alessandro DI GIOVANNI

Events / Meetings

1. Project Vision
2. Release Planning
3. Sprint
4. Sprint Planning
5. Daily Scrum / Daily Stand-up
6. Sprint Review
7. Sprint Retrospective
8. Backlog Grooming / Backlog Refinement
9. Scrum of Scrums o Post-scrum

661

Copyright © Ing. Alessandro DI GIOVANNI

Events: Project Vision

Il ciclo SCRUM inizia con una riunione degli stakeholder, durante la quale:

- ▶ vengono spiegate le esigenze di business che il progetto dovrà soddisfare
- ▶ viene creata la **Vision del Progetto**, accompagnata dal **Budget** e da un **Charter** (che ne definisce obiettivi e finalità)
- ▶ Si fissano a priori **costi e tempi**
- ▶ Ci si focalizza sul problema e non sulla soluzione

Al termine si ottiene l'artefatto **Project Vision Statement**

662

Copyright © Ing. Alessandro DI GIOVANNI

Events: Release Planning

Dopo la Project Vision si organizza il Release Planning Meeting in cui:

- ▶ si deve sviluppare un **piano di rilascio**
- ▶ si identifica il **MVP** (Minimum Viable Product)
- ▶ vengono pianificati i rilasci (release) del prodotto fissando i tempi

663

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint

- ▶ È un'unità di base dello sviluppo in Scrum
- ▶ Rappresenta un blocco rapido di lavoro, alla fine del quale «consegnare una versione al cliente»
- ▶ Durata fissa (1-4 settimane)
- ▶ NO pause
- ▶ Timeboxed
- ▶ Ogni Sprint è preceduto da una riunione di pianificazione (Sprint Planning)
- ▶ Durante uno sprint non è permesso cambiare gli obiettivi (Sprint Goal)
 - ▶ modifiche sospese fino alla successivo Sprint Planning e prese in considerazione nel successivo Sprint
 - ▶ Durante lo sprint il Dev Team può modificare lo Sprint Backlog solo se le modifiche non variano lo Sprint Goal
- ▶ Al termine di ogni sprint il team di sviluppo consegna una **versione potenzialmente completa e funzionante** del prodotto, contenente gli avanzamenti decisi nella riunione di pianificazione dello sprint

664

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Planning

- ▶ Riunione di pianificazione dello Sprint
- ▶ Le US da svolgere vengono prese dal Product Backlog
 - ▶ Sono già stimate, ma la stima può essere rivista sulla base dei maggiori dettagli a disposizione
- ▶ Le US vengono scomposte in Task da eseguire
- ▶ Vi partecipano:
 - ▶ Product Owner
 - ▶ Scrum Master
 - ▶ Dev Team
 - ▶ Key Stakeholder (se necessario)
 - ▶ Manager (se necessario)



665

Copyright © Ing. Alessandro DI GIOVANNI

Events: Daily Scrum

- ▶ Ogni giorno durante lo Sprint
- ▶ Serve a tenere allineato il Dev Team sulla situazione attuale
- ▶ Ciascun membro del Dev Team deve rispondere a 3 domande:
 1. Cosa ho fatto ieri?
 2. Cosa farò oggi?
 3. Vedo ostacoli al lavoro da fare oggi?
- ▶ NO riunione dove il manager raccoglie informazioni sulle attività in ritardo
- ▶ SI incontro dove i membri del team prendono impegni fra loro
- ▶ NO problem solving
- ▶ Chiunque può assistere
- ▶ Possono parlare solo i «pig»: membri del team, SM e PO
- ▶ Scopo: evitare troppi incontri inutili

666

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Review

- ▶ Il Dev Team presenta il lavoro completato (Done) a PO, stakeholder, manager, ...
- ▶ È praticamente la «demo»
- ▶ È informale
- ▶ Si rilascia un software funzionante
 - ▶ Il lavoro incompleto non può essere dimostrato
- ▶ Il PO conferma l'incremento solo se soddisfa i criteri di accettazione concordati
- ▶ NO Slide

667

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective

- ▶ Dev Team + SM ripercorrono lo sprint appena concluso, individuando eventuali elementi di miglioramento per gli sprint successivi
- ▶ Si fa dopo lo Sprint Review
- ▶ Partecipano tutti (SM, PO, Dev Team e eventualmente anche clienti ed utenti ed altri stakeholders)
- ▶ Scopo:
 1. Esaminare come è andato l'ultimo Sprint riguardo a persone, relazioni, processi e strumenti
 2. Identificare e ordinare gli elementi principali che sono andati bene e le migliori potenziali
 3. Creare un piano per attuare i miglioramenti al modo di lavorare del Team
- ▶ Tipicamente si crea una tabella con:
 - ▶ Start doing
 - ▶ Stop doing
 - ▶ Continue doing

La retrospective spesso viene denominata
Start-Stop-Continue Meeting

668

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

Vediamo alcuni metodi per gestire la retrospective...

Fonte: https://dzone.com/articles/11-ideas-to-spice-up-your-retrospective?utm_medium=feed&utm_source=feedpress.me&utm_campaign=Feed%3A+dzone%2Fagile

669

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

1. *The Feedback Game*

The feedback game gives you an insight on your own personality and can be used to start an in-depth conversation about each other's attitude and behavior.

By using 140 cards with different characteristics you'll be able to find an answer to questions like:

- ▶ What are my qualities?
- ▶ How do others perceive me?
- ▶ What are qualities that I want or need to improve?

You can use the game for team building, appraisals, coaching and counseling, career planning, development, and personal strengths.

The game helped in creating a clear image of how a person sees him or herself and how someone else perceives that person.

It also clarifies how well a team can assess qualities of other team members.

670

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

2. Discuss the Team's Purpose

A purpose is the reason for which something is done or created or for which something exists.

A clear, inspiring and ambitious purpose will invite the team to continuously improve themselves.

Without a purpose, the team is at risk to become Scrum zombies that only mechanically attend the Retrospective.

671

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

3. The Constellation Game

Place a sphere in the middle of free space and gather the team around it.

The sphere represents the center of agreement.

Move toward the center if you agree and outwards if you don't.

Read out statements like:

- ▶ I feel I can talk openly in this Retrospective.
- ▶ I am happy with the quality of our code.
- ▶ I feel I'm part of the best team ever.

Afterward, ask which constellations were surprising

672

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

4. Why Retrospectives?

Go back to the roots and start the Retrospective by asking, “why are we having Retrospectives?”

Write down all answers and then post them for everyone to see.

You might be surprised.

673

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

5. Expectations

Each team member gets a sheet of paper that's blank on the lower half and has two sections in the top half:

- ▶ What my teammates can expect from me.
- ▶ What I expect from my teammates.

Participants fill out the top half for themselves.

Afterward, they pass their paper to the left and review the sheet they got.

In the lower half, they write what they personally expect from the author, sign it and pass it on.

After a full round, review and discuss expectations.

674

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

6. *Writing the Unspeakable*

Are there unspoken taboos holding back the team? Stress confidentiality.

Announce the activity as silent and that all evidence will be destroyed afterward.

Hand each participant a sheet of paper to write down the biggest unspoken taboo in the company or team.

When everyone's done, they pass their paper to their left neighbors.

The neighbors read them and may add comments.

Papers are passed on until they return to their authors for a final review.

Then all pages are ceremoniously shredded.

Use with caution!

675

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

7. *Remember the Future*

Introduce this scenario: "Imagine you could time travel to the end of the next iteration. You learn that it was the best, most productive iteration yet! How do your future selves describe it? What do you see and hear?"

Give the team a little time to imagine this and jot down some keywords.

Then let everyone describe their vision of a perfect iteration.

Follow up with "What changes did we implement that resulted in such a productive and satisfying future?"

Write down the answers on cards for the next phase.

676

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

8. *Circle of Influence and Concern*

Prepare a flip chart with three concentric circles, each big enough to put stickies in. Label them from inner to outer circle:

- ▶ Team controls: direct action.
- ▶ Team influences: persuasive action.
- ▶ System: response action.

Sort your insights from the last phase into the circles. In pairs, the participants write down possible actions - preferably addressing issues in their circle of influence.

They post their actions next to the respective issue.

Agree on which plans to try, i.e., by each participant distributing three votes.

677

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

9. *Follow Through*

Let everyone draw an emoticon of their current mood on a sticky note.

Then draw a scale on a flip chart, labeled "Probability we'll implement our action items."

Mark "0%" on the left and "100%" on the right.

Ask everyone to place their sticky according to their level of confidence in their follow through as a team.

Discuss interesting results such as low probability or bad mood.

678

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

10. Choose a Different Location

Choosing a different location for the Retrospective might do wonders.

Leave the office for once.

Choose for example a coffee bar, public park or even a boat as the location.

A completely different environment really helps the team with brainstorming and defining out-of-the-box ideas.

679

Copyright © Ing. Alessandro DI GIOVANNI

Events: Sprint Retrospective - metodi

11. Ask Someone Else to Facilitate the Retrospective

Although the Scrum Master should support the team to continuously improve themselves, this doesn't automatically mean (s)he always the Retrospectives facilitator.

It prevents the Retrospective to become the "Scrum Master Show" and gives you the opportunity to act more as a team member from within.

680

Copyright © Ing. Alessandro DI GIOVANNI

Events: Backlog Grooming (1)

- ▶ Grooming = governare
- ▶ Riunione di «raffinamento» del backlog
- ▶ Fatta durante lo sprint (1 o 2 volte a settimana)
- ▶ Fatta dal PO + Dev Team (non necessariamente tutto)
- ▶ Scopo: assicurare che il backlog contenga:
 - ▶ item appropriati
 - ▶ item prioritizzati correttamente
 - ▶ item in testa al backlog pronti per essere aggiunti al prossimo sprint

681

Copyright © Ing. Alessandro DI GIOVANNI

Events: Backlog Grooming (2)

- ▶ **ATTIVITA'**:
 - ▶ Eliminazione di US non più rilevanti
 - ▶ Creazione di nuove US nate da nuove scoperte
 - ▶ Riassegnazione delle priorità
 - ▶ Stima delle attività non ancora stimate
 - ▶ Correzione di stime precedenti che variano per nuove scoperte
 - ▶ “Splitting” di Epic o US ad alta priorità troppo grosse per essere terminate in uno sprint
- ▶ Per la stima solitamente si usa il metodo del **planning poker**

Scrum Poker					
0	1	2	3	5	8
13	20	40	100	?	☕

682

Copyright © Ing. Alessandro DI GIOVANNI

Events: Scrum of Scrum

- ▶ Si fa solo in caso di grossi progetti (con + scrum team)
- ▶ Consente a gruppi di team di discutere del loro lavoro, concentrandosi su aree di sovrapposizione e d'integrazione
- ▶ Tenuto solitamente dopo il Daily Scrum di ciascun Dev Team
- ▶ Vi partecipa una persona per Dev Team
- ▶ L'agenda è la stessa dei Daily Scrum, più le seguenti quattro domande:
 1. Che cosa ha fatto la tua squadra dal nostro ultimo incontro?
 2. Cosa conterà di realizzare il tuo team prima che ci incontriamo nuovamente?
 3. C'è qualcosa che vi rallenta o vi impedisce di ottenere l'obiettivo?
 4. Siete in procinto di fare qualcosa che possa essere utilizzato da un altro team?

683

Copyright © Ing. Alessandro DI GIOVANNI

DoD (Definition of Done)

- ▶ Anche detta DoR (Definition of Ready)
- ▶ È uno dei concetti fondamentali in Scrum
- ▶ Determina ciò che sarà davvero consegnato dal team al termine di ciascuno sprint
- ▶ Esistono molte checklist differenti, e molte DoD raccomandate...
- ▶ ...non ce n'è una che possa essere applicata da tutti i team del mondo.
- ▶ Si raffina pian piano
 - ▶ Si può suggerire di standardizzare ciò che il team è davvero in grado di ottenere, e migliorare via via lo standard senza adottare prassi che non si è in grado di applicare.
- ▶ È concordata con il PO
- ▶ Esempi:
 - ▶ “sono stati inclusi alcuni test automatici”
 - ▶ “il codice critico è stato sottoposto a riesame”
 - ▶ “tutte le modifiche sono state sviluppate in test-driven mode”
 - ▶ ...

684

Copyright © Ing. Alessandro DI GIOVANNI

Cancellazione di uno sprint

- ▶ Il PO può cancellare uno sprint durante il suo svolgimento se l'obiettivo dello sprint diventa obsoleto
 - ▶ Esempio 1: se sono cambiate le condizioni di mercato
 - ▶ Esempio 2: se l'organizzazione subisce una modifica
- ▶ Di solito non ha senso cancellare uno sprint

685

Copyright © Ing. Alessandro DI GIOVANNI

Problemi tipici da gestire

- ▶ Poche persone parlano nei meeting
- ▶ Ignoranza dei valori agili e di Scrum
- ▶ Prodotto software non testato alla fine dello sprint (cattiva DoD)
- ▶ Product Backlog non pronto all'inizio dello sprint (cattiva definizione di Ready)
- ▶ Mancanza di facilitazione (o cattiva facilitazione)
- ▶ Mancanza di supporto da parte dei manager
- ▶ Mancanza di supporto da parte degli stakeholder
- ▶ Gestione caotica degli scrum di scrums

686

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM vs Kanban

SCRUM

Le iterazioni (sprint) sono sempre *timeboxed*

Per ciascuna iterazione (sprint), il team è **impegnato** su una specifica quantità di lavoro (commitment)

Usa la **rapidità** come metrica di default per la pianificazione ed il miglioramento dei processi

Di norma la scomposizione del lavoro è fatta in modo che ciascuna iterazione possa rilasciare una versione del prodotto idealmente funzionante

Il **backlog** dello sprint è gestito da un unico team

La composizione del backlog è guidata dalle **priorità**

Sono definiti tre ruoli principali (Product owner, Scrum Master, Team)

KANBAN

Le iterazioni non sono sempre *timeboxed*

Il commitment è opzionale

Usa il **lead time** come metrica di default per la pianificazione ed il miglioramento dei processi

Nessuna prescrizione sulla scomposizione del lavoro

Il **kanban board** può essere condiviso tra più team e singoli individui

L'assegnazione delle priorità è opzionale

Non sono definiti ruoli particolari

687

Copyright © Ing. Alessandro DI GIOVANNI

SCRUM in meno di 100 parole

È un **modello di processo** per produrre software ottenendo il massimo valore utile nel minor tempo

Permette al cliente di ispezionare rapidamente e ripetutamente (ogni 3-4 settimane) versioni funzionanti del software

Il cliente definisce funzioni da realizzare e loro priorità. Il team di sviluppo decide quotidianamente il modo migliore di produrre le funzioni di più alta priorità.

Ogni 3-4 settimane nasce una nuova versione che viene esaminata per decidere se continuarne lo sviluppo con un altro sprint o produrre un rilascio

688

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (1)

- ▶ “The Machine That Changed World”
 - ▶ E’ il libro da cui partire!
 - ▶ In questo libro non si parla di Scrum, ma si racconta la storia della nascita del TPS (Toyota Production System), della contrapposizione del modello tayloristico della produzione industriale di massa con il modello “leggero” adottato e raffinato in Toyota a partire dal secondo dopoguerra.
 - ▶ È il libro per chi è interessato a conoscere la storia che potremmo dire essere la spiegazione di tutto quello che è successo dopo.
 - ▶ Concetti di Agile e Lean appariranno con una visione diversa dopo aver letto questo libro.

689

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (2)

- ▶ “Agile Samurai”
 - ▶ consigliato a chi, dopo aver letto la storia del TPS, volesse approfondire meglio i concetti di Agile in un contesto più inerente allo sviluppo del software.
- ▶ “Agile Transition” di Andrea Tomasini e Martin Kearns
 - ▶ Per chi fosse interessato a capire come mettere in pratica i concetti di agile all’interno della propria azienda o del proprio team.
 - ▶ Il libro è pubblicato su iBooks Store di Apple, ma si trova in rete anche una versione PDF.

690

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (3)

- ▶ "Guida Scrum"
 - ▶ testo di riferimento
 - ▶ scaricabile dal sito ufficiale
- ▶ "AgileAtlas"
 - ▶ piccolo ma essenziale testo che racconta in 10 pagine tutto quello che c'è da sapere nella metodologia di Scrum
 - ▶ Curato e distribuito da Scrum Alliance

691

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (4)

- ▶ "Scrum and XP from the trenches" ("Scrum e XP dalle trincee")
 - ▶ testo di riferimento scritto da uno dei padri di Scrum.
 - ▶ Importante testimonianza utile a molti.
 - ▶ Lettura semplice
 - ▶ mette enfasi sul concetto che Scrum prima di tutto è un framework basato su principi
 - ▶ Offre una visione su come gli autori hanno messo in pratica Scrum nel loro team di progetto.

692

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (5)

- ▶ “Scrum Field Guide”
 - ▶ efficace, ben fatto e si legge facilmente.
 - ▶ Alcuni passaggi sono certamente utili come ad esempio la descrizione della giornata tipo del PO o dello SM.
- ▶ “Essential Scrum”
 - ▶ dice cose utili sia per passare l’esame di certificazione, sia nel lavoro quotidiano di coach agile in azienda.
- ▶ “Kanban and Scrum - Ottenere il massimo da entrambe”
 - ▶ Per chi fosse interessato ad approfondire il tema del confronto fra Kanban e Scrum

693

Copyright © Ing. Alessandro DI GIOVANNI

Lecture consigliate (6)

- ▶ “Liftoff” di Diana Larsen
 - ▶ Parla di LiftOff (“decollo”), cioè la fase iniziale di un progetto, fase che prevede la definizione della vision e il relativo processo di trasformazione da idea astratta a elenco concreto di cose da fare. Per questo si parla spesso di “From Vision to Backlog (and back)”.
- ▶ “Agile Retrospectives” di Diana Larsen ed Easter Derby
 - ▶ Tratta il tema della retrospettiva agile
- ▶ “User Stories Applied”
 - ▶ un testo per approfondire il tema della gestione delle storie
 - ▶ Propone alcuni concetti un po’ datati, ma utili per neofiti

694

Copyright © Ing. Alessandro DI GIOVANNI

Letture consigliate (7)

- ▶ "The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity"
 - ▶ sulla creazione di storie tramite la user story mapping e le personas

695

Copyright © Ing. Alessandro DI GIOVANNI

Un buon approccio AGILE?

Scrum x pianificazione

Kanban x comunicazione

XP x controllo

696

Copyright © Ing. Alessandro DI GIOVANNI