

Sviluppo software Agile

Cesare Gallotti

2017-01-30

Cesare Gallotti



Opera rilasciata sotto la Creative Commons Attribuzione 4.0 Internazionale (<http://creativecommons.org/licenses/by/4.0/deed.it>).
Bisogna attribuire il lavoro a Cesare Gallotti con link a <http://www.cesaregallotti.it>

Agenda

- Agile in breve
 - Cos'è Agile
 - Persone, eventi, *artifact*
 - Strumenti
- Agile e le certificazioni ISO

Agile in breve

Cos'è Agile

- L'approccio Agile nasce negli anni Novanta dopo aver analizzato i problemi dell'approccio tradizionale "a cascata" o "waterfall".
- I principi seguiti sono riportati nel "Manifesto per lo Sviluppo Agile di Software":
 - «Gli individui e le interazioni più che i processi e gli strumenti
 - Il software funzionante più che la documentazione esaustiva
 - La collaborazione col cliente più che la negoziazione dei contratti
 - Rispondere al cambiamento più che seguire un piano
 - Ovvero, fermo restando il valore delle voci a destra, consideriamo più importanti le voci a sinistra».
- I metodi "Agili" sono diversi (Scrum, Extreme programming, DSDM, Crystal clear, Lean, ecc); ciascuno può concentrarsi su un aspetto dello sviluppo (gestione del progetto, processi tecnici, pratiche).
 - Qui si farà riferimento soprattutto allo schema "Scrum", molto diffuso in Italia e concentrato su come gestire i progetti.
 - Vedere "The Definitive Guide to Scrum: The Rules of the Game", developed and sustained by Ken Schwaber and Jeff Sutherland.

Approccio waterfall

- Problemi dell'approccio waterfall:
 - ambito (scope) definito completamente all'inizio, quando in realtà non si conoscono tutte le esigenze;
 - ogni richiesta di cambiamento può comportare costi elevati;
 - per progetti lunghi e complessi, i requisiti iniziali non sono più quelli finali (cambiamento delle esigenze nel tempo);
 - problemi di disallineamento tra aspettative degli utenti e prodotto finale;
 - il prodotto finale ha molte funzionalità poi non utilizzate;
 - i tempi e i costi e la qualità sono difficili da rispettare.

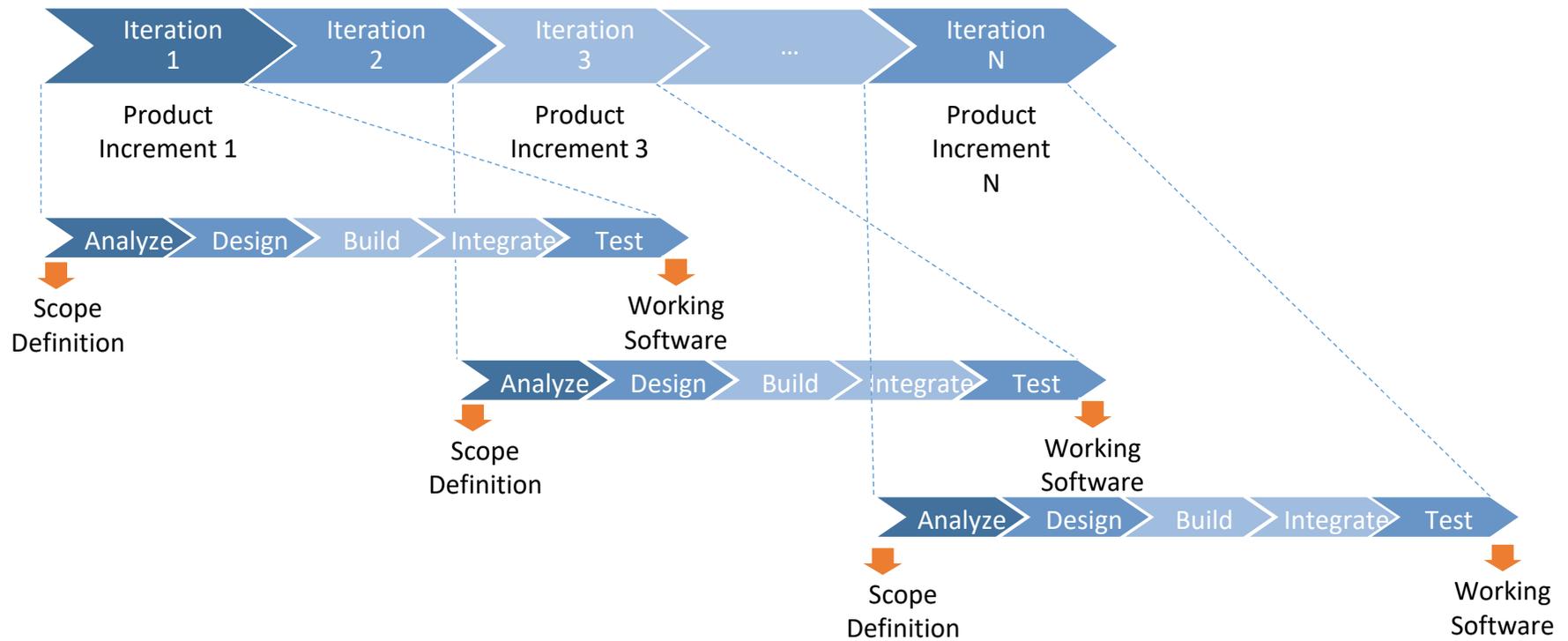


Approccio Agile (1/3)

- Caratteristiche dell'approccio Agile:
 - non definire completamente i requisiti all'inizio, ma man mano, quando sono utili o prioritari;
 - presentare frequentemente al cliente gli avanzamenti funzionanti;
 - riesaminare e scegliere frequentemente i requisiti prioritari;
 - il cliente e i suoi utenti devono essere coinvolti frequentemente.
- Operativamente:
 - stabilire eventi di tipo "timeboxed";
 - stabilire "Iteration" di 2-4 settimane, dette "sprint";
 - a inizio sprint stabilire cosa sarà fatto (nessun cambiamento consentito);
 - a fine sprint riesaminare tutto con il cliente e gli utenti;
 - a fine sprint valutare i possibili miglioramenti di processo.



Approccio Agile (2/3)



Approccio Agile (3/3) – I 12 principi

1. Soddisfazione dei clienti;
2. Accogliamo i cambiamenti nei requisiti;
3. Consegnamo frequentemente software funzionante;
4. Committenti e sviluppatori devono lavorare insieme;
5. Fondiamo i progetti su individui motivati e diamo loro fiducia;
6. Priilegiamo conversazioni faccia a faccia;
7. Il software funzionante è il principale metro di misura di progresso;
8. Gli sponsor, gli sviluppatori e gli utenti mantengono un ritmo costante;
9. Continua attenzione all'eccellenza tecnica e alla buona progettazione;
10. La semplicità è essenziale;
11. Team che si auto-organizzano;
12. Il team riflette su come diventare più efficace.

Quali sono le componenti di Scrum

1. Persone (Scrum team):
 - Product owner;
 - Scrum master;
 - Development team;
2. Eventi
 - The Sprint;
 - Sprint Planning;
 - Daily Scrum;
 - Sprint Review;
 - Sprint Retrospective;
3. Scrum Artifacts
 - Product Backlog
 - Sprint Backlog
 - Increment
 - Definition of “Done”

Scrum team (1/2)

- Product owner (uno solo per scrum team):
 - è l'unico riferimento del cliente;
 - stabilisce i requisiti funzionali (Product Backlog items);
 - li ordina per priorità;
 - li rende visibili;
 - stabilisce le date per le release.
- Scrum master (uno solo per scrum team):
 - assicura che il processo sia seguito (non è un project manager);
 - “protegge” il Development team.

Scrum team (2/2)

- Development Team (3-9 persone):
 - stabilisce autonomamente lo sforzo richiesto per ogni attività;
 - si auto organizza (lo Scrum master è esterno!);
 - comprendono tutte le competenze necessarie;
 - non hanno titoli specifici;
 - per progetti più grandi, si hanno più Development team.

Eventi (1/2)

- Sprint:
 - ha durata fissa (2-4 settimane);
 - comprende Sprint Planning, Daily Scrums, lo sviluppo e test, Sprint Review e Sprint Retrospective;
 - durante lo sprint non si cambiano i requisiti;
 - ogni sprint è un progetto completo.
- Sprint planning:
 - dal Product backlog, il Development team stima lo sforzo necessario per realizzare le User stories più prioritarie;
 - dal Product backlog, il Development team estrae le User stories che possono essere realizzate durante lo sprint, considerandone priorità e richiesta di risorse; crea così lo Sprint backlog.

Eventi (2/2)

- Daily Scrum o Daily stand-up (15 minuti):
 - riunione quotidiana del Development team (altri possono assistere).
- Sprint review (2-4 ore):
 - alla fine dello sprint;
 - partecipano tutte le parti interessate (clienti, utenti, ecc.) e lo Scrum Team;
 - il Development team illustra quanto fatto (usando software funzionante, non presentazioni);
 - si discute su cosa fare in futuro (output è un Product backlog aggiornato).
- Sprint retrospective (1-3 ore)
 - partecipa lo Scrum team;
 - alla fine dello sprint;
 - si analizza come si è svolto lo Sprint appena concluso;
 - si riesamina la "Definition of done";
 - si stabiliscono i possibili miglioramenti.

Artifact (1/2)

- Product Backlog (con Product backlog item o PBI):
 - contiene le “user stories”, requisiti funzionali visti con l’ottica degli utilizzatori del software;
 - le User stories sono ordinate in priorità sulla base di
 - rischi;
 - dipendenze;
 - scadenze;
 - valore per il cliente;
 - il Product backlog non è statico;
 - User stories meno prioritarie sono spesso accorpate in “Epiche” (di bassa priorità), poi raffinate quando diventano prioritarie;
 - le User stories sono collegate a “Task” o “attività”;
 - il Development team stima le User stories, non i Task.

Artifact (2/2)

- Sprint Backlog:
 - le Storie da realizzare in un determinato Sprint;
 - lo Sprint backlog si fa solo a inizio Sprint (non si pianificano gli Sprint successivi).
- Increment:
 - quanto consegnato dal Development team a fine Sprint;
 - il nuovo incremento deve essere "Done";
 - si raccomanda di operare in "continuous integration", in modo che ogni piccolo cambiamento sia subito integrato con il prodotto e oggetto di compilazioni e test frequenti.
- Definition of "Done":
 - la stabilisce lo Scrum Team (allineata alle regole aziendali);
 - può includere i test (tecnici, di integrazione, di accettazione; in diversi ambienti), la documentazione.

Strumenti

- Per gestire le attività Scrum sono disponibili strumenti gestionali (il più diffuso in Italia è Jira, ma a rigore potrebbero essere usati anche tabelloni con etichette) per la gestione di:
 - user stories (con le loro caratteristiche), inclusi i bug;
 - task (con le loro caratteristiche, inclusi quelli riaperti);
 - product e sprint backlog;
 - stato di avanzamento del prodotto e dello sprint;
 - ulteriori scadenze.
- Per le attività tecniche sono disponibili strumenti per:
 - la configurazione e il versionamento del software;
 - la continuous integration;
 - il controllo del codice;
 - i passaggi in ambiente di produzione.

Altro

- Agile usa termini vari. Per esempio:
 - grooming;
 - scrum di scrum.
- Attenzione che normalmente ci troviamo di fronte a impostazioni “Agile but”.

Agile e le certificazioni ISO

Agile e qualità

- Pianificazione della progettazione:
 - è il metodo stesso (può essere documentato in una pagina);
 - product backlog e sprint backlog.
- Riesami e verifiche:
 - sprint review (lista partecipanti, nuovi PBI; lista dei PBI conclusi);
 - sprint retrospective (lista partecipanti, miglioramenti pianificati).
- Validazioni:
 - Definition of Done;
 - log di ciascuna User story o di ciascun Task (“Test fatto da.... il....”);
 - sprint review (lista dei PBI conclusi).
- Modifiche:
 - product backlog;
 - eventuali Task riaperti.

Agile e sicurezza delle informazioni

- Requisiti di sicurezza:
 - la cosa più difficile da prevedere; si potrebbero prevedere dei PBI con flag di sicurezza o una documentazione complessiva dell'architettura di sicurezza.
- Politica di sviluppo sicuro:
 - nulla la vieta in Agile.
- Gestione dei cambiamenti:
 - le User stories o i Task possono avere degli attributi (p.e. il livello di rischio).
- Test di sicurezza:
 - possono essere inclusi nella Definition of Done;
 - possono anche essere effettuati dalle Operations e quindi non essere in carico allo Scrum Team.