

Agile in a Complex Environment

Project Management Forum

Lugano, Switzerland

June 22-23, 2017

Giri Sapthagirinath

PMP, Scrum Master, Program Manager

sapthagirinath@gmail.com

Nora Sleumer

PMP, Scrum Master, Scaled Agile Consultant

nora.sleumer@nelcom.ch

Introduzione

- La metodologia agile funziona al meglio con piccoli team coibitati in cui il software può essere impiegato per una produzione rapida e non è richiesta una documentazione completa dei processi di gestione dei progetti.
- In una grande società che gestisce progetti complessi solitamente le cose non funzionano così.
- Ciononostante, il management ha riconosciuto che la metodologia agile offre risultati migliori rispetto al tradizionale modello a cascata.
- Vedremo com'è possibile integrare il metodo agile in un ambiente di lavoro complesso.

Quali sono alcuni aspetti di Agile?

- La metodologia agile può avere diversi significati per i vari stakeholder in una società:
- **Per il management**, agile può significare un tasso più elevato di progetti riusciti, visto che in passato molti progetti a cascata tradizionali non hanno funzionato
- **Per i clienti**, il modello agile può implicare la capacità di cambiare idea in qualsiasi momento pur ottenendo il risultato auspicato alla data prevista e senza costi aggiuntivi
- **Per il team di sviluppo**, la minore enfasi sulla documentazione può rappresentare un grande vantaggio, mentre per il responsabile di progetto il lavoro è meno complesso, in quanto il team si organizza in maniera autonoma

Esigenze di un'azienda complessa

- Per quanto un'organizzazione possa essere attratta dai benefici del "miracolo" agile, in un contesto complesso esistono fattori che devono essere presi in considerazione.
- Una grande organizzazione richiede:
 - pianificazione delle risorse,
 - previsioni di budget,
 - ruoli predefiniti e
 - PSR (project status report, rapporti sullo stato del progetto) mensili, bisettimanali e settimanali.
 - Per il deployment di una nuova versione possono essere necessari mesi di test prima che vi siano prove sufficienti che la produzione non verrà interrotta.

Manifesto per lo sviluppo agile di software

- Manifesto per lo sviluppo agile di software: siamo alla ricerca di modi migliori di creare software, sperimentandoli e aiutando gli altri a fare lo stesso. Grazie a questa attività siamo arrivati a considerare importanti:

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
la risposta al cambiamento più che l'esecuzione di un piano.

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

Scrum

- Per trasformare il manifesto in un processo concreto si può utilizzare il metodo Scrum, ideato da Jeff Sutherland (<https://www.scruminc.com/>) e Ken Schwaber (<https://www.scrum.org/>) nel 1995.
- Sutherland avrebbe affermato che i tre fattori di distinzione tra i team Scrum e i team tradizionali sono:
 - gestione autonoma
 - continuità nella composizione del team
 - impegno nei singoli progetti.



Extreme Programming

- Nel campo dello sviluppo di software, è possibile passare a un processo tecnico successivo con l'eXtreme Programming (<http://www.extremeprogramming.org/>) reso popolare da Kent Beck.
- Alcuni dei cardini su cui si fonda XP sono:
 - scrittura di un test automatico, seguita dallo sviluppo del codice
 - integrazione frequente del codice nel code branch principale, dopo essersi assicurati che tutti i test siano andati a buon fine
 - misurazione della velocità del progetto.

My background with XP & Agile

1. **XP** 7 PT per release ● weekly releases ● 8000 autom. tests



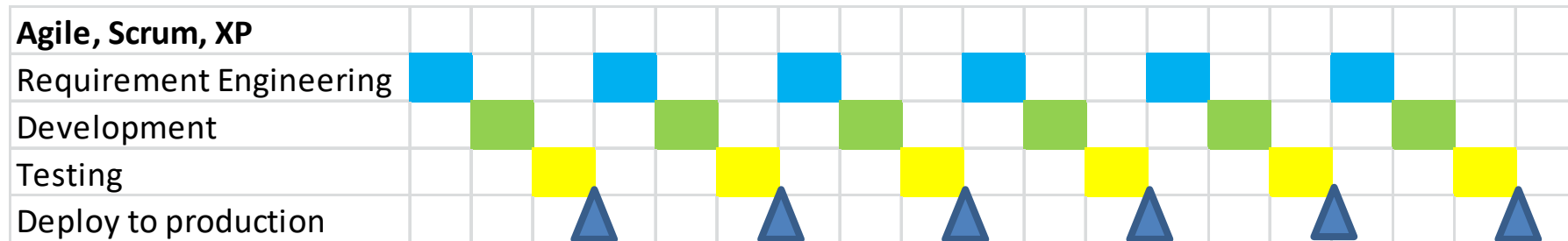
2. **N-** 700 PT per release ● every 3 months ● no autom. tests



3. **N+** 7000 PT per release ● 1 rel. after 3 years ● no autom. tests



Agile, Scrum, Extreme Programming



- In ogni Sprint si fa
 - Requirement engineering
 - Software development
 - Testing
 - Deployment
- Tutto con le stesse persone

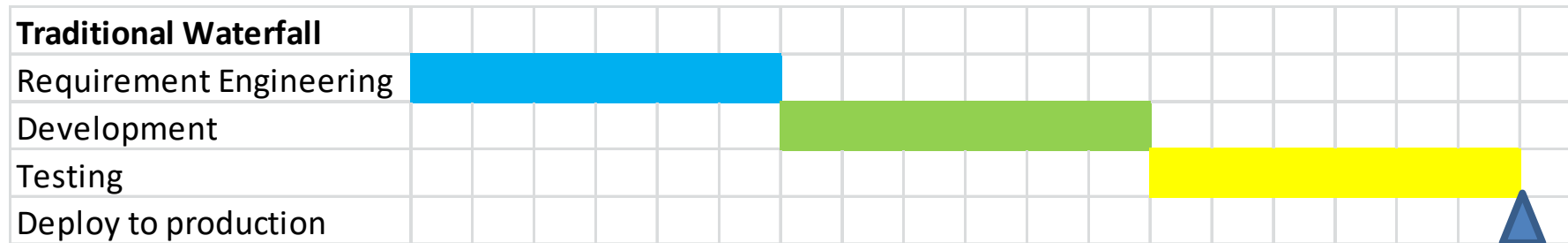
Vantaggi del modello agile

- Coinvolgimento attivo di tutti gli stakeholder. Interazioni quotidiane tra clienti (utenti/business) e sviluppatori.
- Costante attenzione all'eccellenza tecnica e alla progettazione.
- Maggior focus su prodotti praticabili.
- Regolare adeguamento alle mutevoli circostanze.
- Clienti, sviluppatori e responsabili dei test interagiscono costantemente tra di loro, assicurando una buona comunicazione.
- Maggiore soddisfazione del cliente grazie alla fornitura rapida e continua di software utile.

Svantaggi del modello agile

- Per i progetti piuttosto grandi e complessi, è difficile valutare il lavoro necessario per l'intera portata all'inizio dello sviluppo del software.
- La minore enfasi sulla progettazione e sulla documentazione può portare a informazioni insufficienti per la manutenzione del software.
- Il progetto può deragliare facilmente se il rappresentante del cliente non ha le idee chiare sull'esito finale auspicato.
- Sono assolutamente necessarie risorse competenti ed esperte. Non vi è tempo per una formazione on the job.
- Vi è una situazione di dipendenza dalla disponibilità del rappresentante del cliente in tempo reale.
- Il cliente può credere di essere in grado di apportare modifiche in qualsiasi momento senza alcun impatto sul costo o sulla tempistica.

Cascata - Waterfall



- Con la tradizionale Waterfall / Cascata
 - Un team fa il requirement engineering
 - Un altro team sviluppa
 - Ancora altre persone testano
 - Un team separato lo mette in produzione

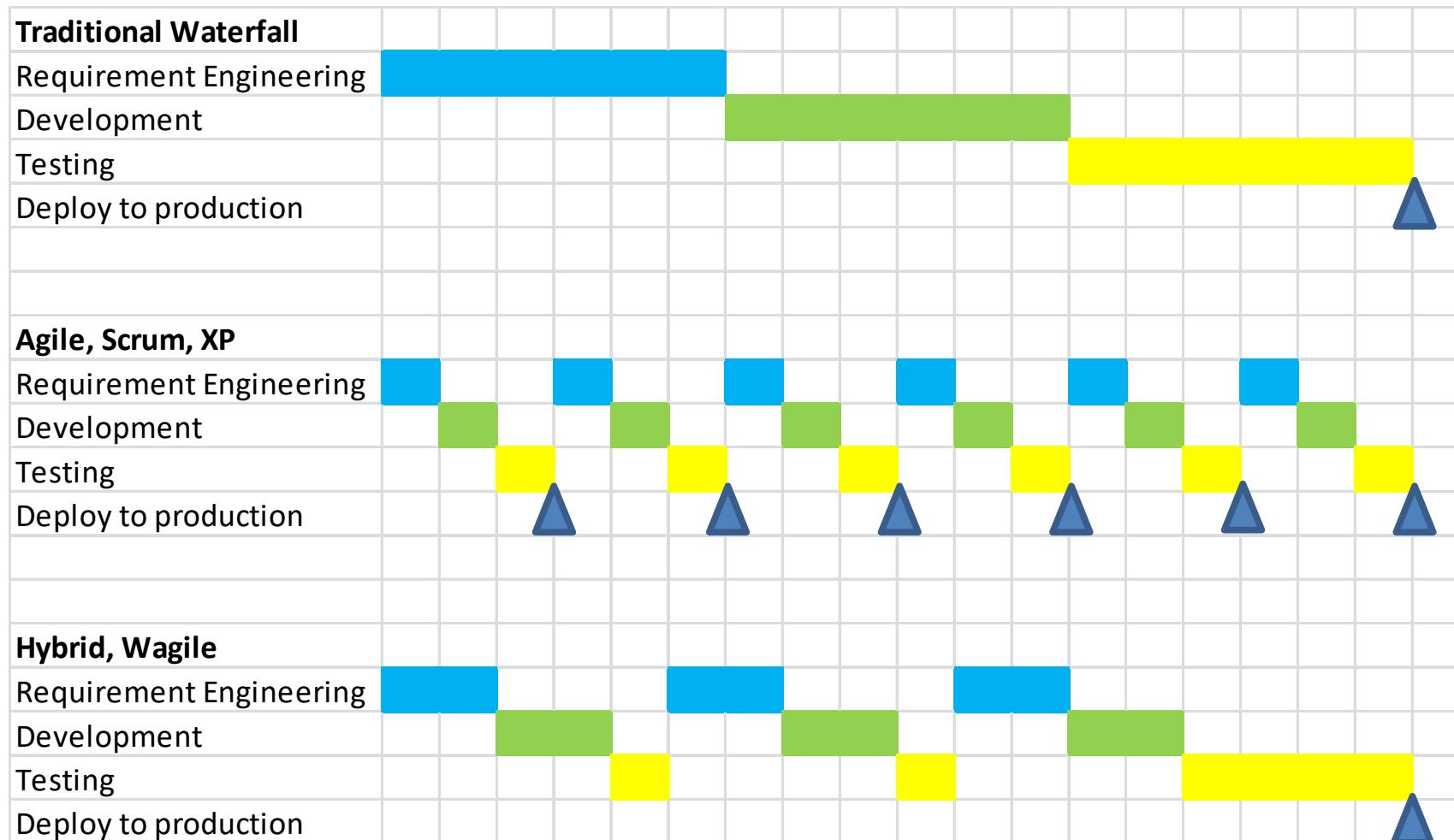
Vantaggi del modello Waterfall

- Logica lineare e consequenziale.
- Maggiore controllo presunto sul progetto.
- Ogni fase può essere assegnata a team separati.
- Applicazione di un'opportuna disciplina nei vari cicli di sviluppo predisposti: ogni fase ha un punto di inizio e una fine ben definiti.
- Le scadenze possono essere fissate per ogni fase, assicurando, in teoria, che il progetto venga completato nel rispetto dei tempi previsti.
- C'è tempo per testare in modo completo l'insieme dei requisiti.

Svantaggi del modello Waterfall

- La natura lineare e consequenziale del processo presuppone che il cliente sia in grado di indicare esplicitamente tutti i requisiti sin dalle fasi iniziali del ciclo di vita. Ciò non risulta conveniente per progetti nei quali vi è un rischio da moderato a elevato che i requisiti possano cambiare.
- È difficile stimare tempi e costi a uno stadio di sviluppo iniziale e, pertanto, i progetti sono soggetti a modifica man mano che si prosegue lungo il loro ciclo di vita.
- Il software operativo/le funzionalità sono disponibili per il cliente solo alla fine del ciclo di vita del progetto.

Wagile: un approccio ibrido



Wagile: un approccio ibrido

- Il tradizionale approccio a waterfall prevede tre (o più) chiare fasi: ingegneria/analisi dei requisiti, sviluppo/implementazione e testing/garanzia di qualità
- L'approccio agile raggruppa tutte e tre le fasi in un unico "sprint" o ciclo breve e rilascia il codice per la produzione (deployment).
- Nell'approccio ibrido o wagile, i requisiti sono raggruppati in sprint più lunghi, che vengono analizzati, implementati e testati in un ambiente locale
- Dopo un paio di grandi sprint tutto il codice viene testato in un ambiente di prova del sistema e rilasciato per la produzione
- La complessità è scomposta in sezioni più facilmente gestibili



Wagile: un approccio ibrido

- La pianificazione può essere effettuata per tutta la gamma di requisiti fino al deployment, soddisfacendo quindi le esigenze di pianificazione a lungo termine del management.
- Se i vari ruoli sono chiaramente differenziati possono già iniziare ad analizzare la serie successiva di requisiti accelerando quindi il processo.
- Durante il ciclo di vita vi è più tempo a disposizione per soddisfare le esigenze di documentazione di un'organizzazione complessa.
- Vi è tempo anche per affinare i requisiti per gli ultimi cicli di implementazione mentre il lavoro è già in corso nel primo ciclo. Questo aumenta la fiducia nei risultati del team, concedendo all'azienda anche il tempo di abituarsi ai requisiti ove necessario.
- I test a livello di sistema sono effettuati a un ritmo sostenibile: non in ogni sprint, ma ogni 3 o 4 sprint.

Conclusione

- Ciascuna società ha esigenze e restrizioni specifiche.
- Alcuni ambienti risultano più adatti a un approccio agile; altre aziende possono avere molti più requisiti burocratici che rendono impossibile applicare la metodologia agile nella sua forma più pura.
- Abbiamo presentato un approccio ibrido o wagile in grado di generare i vantaggi tipici della metodologia agile pur rispondendo alle esigenze di ambienti più complessi.

