Processo di sviluppo SCRUM

Ken Schwaber

Metodi di sviluppo avanzati 131
Middlesex Turnpike Burlington, MA 01803 email
virman@aol.com Fax: (617) 272-0555

ASTRATTO. *La filosofia dichiarata e accettata per lo sviluppo dei sistemi è che il processo di sviluppo è un approccio ben compreso che può essere pianificato, stimato e completato con successo. Ciò si è rivelato errato nella pratica. SCRUM presuppone che il processo di sviluppo dei sistemi sia un processo imprevedibile e complicato che può essere descritto solo approssimativamente come una progressione complessiva. SCRUM definisce il processo di sviluppo dei sistemi come un insieme ampio di attività che combina strumenti e tecniche conosciuti e utilizzabili con il meglio che un team di sviluppo può escogitare per costruire sistemi. Poiché queste attività sono libere, vengono utilizzati controlli per gestire il processo e il rischio intrinseco. SCRUM è un miglioramento del ciclo di sviluppo orientato agli oggetti iterativo/ incrementale comunemente utilizzato.*

PAROLE CHIAVE: *SCRUM SEI Capacità-Maturità-Modello Processo Empirico*

---

1. Introduzione

In questo documento introduciamo un processo di sviluppo, SCRUM, che tratta le parti principali dello sviluppo dei sistemi come una scatola nera controllata. Mettiamo in relazione questo con la teoria della complessità per mostrare perché questo approccio aumenta la flessibilità e produce un sistema che risponde sia ai requisiti iniziali che a quelli aggiuntivi scoperti durante lo sviluppo in corso.

Sono stati tentati numerosi approcci per migliorare il processo di sviluppo dei sistemi.
Ciascuno è stato pubblicizzato come in grado di fornire "significativi miglioramenti della produttività". Tutti non sono riusciti a produrre miglioramenti drammatici.1 Come ha osservato Grady Booch, "Spesso chiamiamo questa condizione crisi del software, ma francamente, una malattia che si protrae da così tanto tempo deve essere definita normale".[2]

In questo documento i concetti del controllo dei processi industriali vengono applicati al campo dello sviluppo di sistemi. Il controllo dei processi industriali definisce i processi come "teorici" (completamente definiti) o "empirici" (scatola nera). Quando un processo di scatola nera viene trattato come un processo completamente

---

[1] Brooks, FP "Nessuna soluzione miracolosa: l'essenza e gli incidenti dell'ingegneria del software." Computer 20:4:10-19, aprile 1987.

[2] Analisi orientata agli oggetti e progettazione con applicazioni, p. 8, Grady Booch, Benjamin/Cummings Società editrice, Inc., 1994

processo definito, si verificano risultati imprevedibili. Un'ulteriore trattazione di questo aspetto è fornita nell'Appendice 1.

Un numero significativo di processi di sviluppo dei sistemi non sono completamente definiti, ma vengono trattati come se lo fossero. Imprevedibilità senza risultati di controllo. L'approccio SCRUM tratta questi processi di sviluppo dei sistemi come una scatola nera controllata.

Varianti dell'approccio SCRUM per lo sviluppo di nuovi prodotti con piccoli team ad alte prestazioni sono stati osservati per la prima volta da Takeuchi e Nonaka3 presso Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox e Hewlett-Packard. Coplien4 ha osservato che un approccio simile applicato allo sviluppo software presso Borland è il progetto di sviluppo C++ con la produttività più elevata mai documentato. Più recentemente, un approccio perfezionato al processo SCRUM è stato applicato da Sutherland5 allo sviluppo di Smalltalk e Schwaber6 allo sviluppo di Delphi.

L'approccio SCRUM viene utilizzato con notevole successo da aziende di software all'avanguardia. Gli analisti del settore ritengono che SCRUM possa essere appropriato per altre organizzazioni di sviluppo software per ottenere i vantaggi attesi dalle tecniche e dagli strumenti orientati agli oggetti.7

2. Panoramica

Il nostro nuovo approccio allo sviluppo dei sistemi si basa sia sulla gestione dei processi definita che su quella della scatola nera. Chiamiamo questo approccio metodologia SCRUM (vedi Takeuchi e Nonaka, 1986), dal nome della SCRUM nel rugby - una formazione compatta di attaccanti che si uniscono insieme in posizioni specifiche quando viene chiamata una mischia.8

Come verrà discusso più avanti, SCRUM è un miglioramento dell'approccio iterativo e incrementale alla fornitura di software orientato agli oggetti inizialmente documentato da Pittman9 e successivamente ampliato da Booch.10 Potrebbe utilizzare gli stessi ruoli per il personale di progetto delineati da Graham11, ad esempio , ma organizza e gestisce il processo del team in un modo nuovo.

---

[3] Takeuchi, Hirotaka e Nonaka, Ikujiro. Gennaio-Febbraio 1986. “Il Nuovo Sviluppo del Nuovo Prodotto Gioco." Revisione aziendale di Harvard. 4

[4] Coplien, J. "Artigianato del software Borland: un nuovo sguardo a processo, qualità e produttività". Atti della quinta conferenza annuale Borland International, 5 giugno 1994. Orlando, Florida.

[5] Sutherland, Jeff. Home Page di ScrumWeb: una guida al processo di sviluppo di SCRUM. Pagina Web sulla tecnologia degli oggetti di Jeff Sutherland, 1996 <http://www.tiac.net/users/jsuth/scrum/index.html>

[6] Schwaber, Ken. "Caos controllato: vivere al limite." Programmatore americano, aprile 1996.

[7] Gruppo Aberdeen. Aggiornamento alla metodologia ISV per lo sviluppo di applicazioni aziendali. Punto di vista del prodotto 8:17, 7 dicembre 1995.

[8] Gartner, Lisa. Il Primer per Principianti. Radcliffe Rugby Football Club, 1996 <http://vail.al.arizona.edu/rugby/rad/rookie_primer.html>

[9] Pittmann, Matteo. Lezioni apprese nella gestione dello sviluppo orientato agli oggetti. IEEE Software, gennaio 1993, pagine 43-53.

[10] Booch, Grady. Soluzioni a oggetti: gestire il progetto orientato agli oggetti. Addison-Wesley, 1995.

[11] Graham, Ian. Migrazione alla tecnologia a oggetti. Addison-Wesley, 1994.

SCRUM è una metodologia di gestione, miglioramento e manutenzione di un sistema esistente o di un prototipo di produzione. Presuppone la progettazione e il codice esistenti, il che è praticamente sempre il caso nello sviluppo orientato agli oggetti a causa della presenza di librerie di classi.
SCRUM affronterà in un secondo momento gli sforzi di sviluppo di sistemi legacy totalmente nuovi o riprogettati.

I rilasci dei prodotti software sono pianificati in base alle seguenti variabili:

- Requisiti del cliente: come è necessario migliorare il sistema attuale.
- Pressione temporale: quale periodo di tempo è necessario per ottenere un vantaggio competitivo.
- Competizione: cosa sta facendo la concorrenza e cosa è necessario per superarla.
- Qualità - Qual è la qualità richiesta, date le variabili di cui sopra.
- Visione: quali cambiamenti sono necessari in questa fase per realizzare la visione del sistema.
- Risorsa: quale personale e finanziamenti sono disponibili.

Queste variabili costituiscono il piano iniziale per un progetto di miglioramento del software. Tuttavia, queste variabili cambiano anche durante il progetto. Una metodologia di sviluppo di successo deve tenere conto di queste variabili e della loro natura evolutiva.

3. Situazione attuale dello sviluppo

I sistemi sono sviluppati in un ambiente altamente complicato. La complessità è presente sia nell'ambiente di sviluppo che nell'ambiente di destinazione. Ad esempio, quando è stato avviato lo sviluppo del sistema di controllo del traffico aereo, non è stato necessario prendere in considerazione i sistemi client-server a tre livelli e la deregolamentazione delle compagnie aeree. Tuttavia, questi cambiamenti ambientali e tecnici si sono verificati durante il progetto e dovevano essere presi in considerazione nell'ambito del sistema in costruzione.

Le variabili ambientali includono:

- Disponibilità di professionisti qualificati: quanto più nuovi sono la tecnologia, gli strumenti, i metodi e il dominio, tanto più piccolo è il pool di professionisti qualificati.
- Stabilità della tecnologia di implementazione: più nuova è la tecnologia, minore è la stabilità e maggiore è la necessità di bilanciare la tecnologia con altre tecnologie e procedure manuali.
- Stabilità e potenza degli strumenti: più nuovo e potente è lo strumento di sviluppo, minore è il pool di professionisti qualificati e più instabile è la funzionalità dello strumento.
- Efficacia dei metodi: quali metodi di modellazione, test, controllo della versione e progettazione verranno utilizzati e quanto saranno efficaci, efficienti e comprovati.

Competenza nel settore: sono professionisti qualificati disponibili nei vari settori, inclusi affari e tecnologia.

Nuove funzionalità: quali funzionalità completamente nuove verranno aggiunte e in che misura queste si adatteranno alle funzionalità attuali.

Metodologia: l'approccio generale allo sviluppo dei sistemi e all'utilizzo dei metodi selezionati promuove la flessibilità oppure si tratta di un approccio rigido e dettagliato che limita la flessibilità.

Concorso: cosa farà il concorso durante il progetto? Quali nuove funzionalità verranno annunciate o rilasciate.

Tempo/finanziamenti: quanto tempo è disponibile inizialmente e man mano che il progetto avanza? Quanti fondi per lo sviluppo sono disponibili.

Altre variabili: qualsiasi altro fattore a cui è necessario rispondere durante il progetto per garantire il successo del sistema risultante e consegnato, come le riorganizzazioni.

La complessità complessiva è una funzione di queste variabili:

complessità = f(variabili dell'ambiente di sviluppo + variabili dell'ambiente di destinazione)

dove queste variabili possono cambiare e cambiano durante il corso del progetto.

All'aumentare della complessità del progetto, maggiore è la necessità di controlli, in particolare della valutazione continua e della risposta al rischio.

I tentativi di modellare questo processo di sviluppo hanno riscontrato i seguenti problemi:

Molti dei processi di sviluppo sono incontrollati. Gli input e gli output sono sconosciuti o vagamente definiti, il processo di trasformazione manca della necessaria precisione e il controllo di qualità non è definito. I processi di test ne sono un esempio.

Un numero imprecisato di processi di sviluppo che collegano processi noti ma non controllati non è identificato. I processi dettagliati per garantire che un modello logico contenga contenuti adeguati per portare a un modello fisico di successo sono uno di questi processi.

Gli input ambientali (requisiti) possono essere presi in considerazione solo all'inizio del processo. Successivamente sono necessarie complesse procedure di gestione delle modifiche.

I tentativi di imporre un modello metodologico micro o dettagliato sul processo di sviluppo non hanno funzionato perché il processo di sviluppo non è ancora completamente definito. ~~Recitazione~~

come se il processo di sviluppo fosse definito e prevedibile portasse ad essere impreparati di fronte a risultati imprevedibili.

Sebbene il processo di sviluppo sia definito e dinamico in modo incompleto, numerose organizzazioni hanno sviluppato metodologie di sviluppo dettagliate che includono metodi di sviluppo attuali (strutturato, OO, ecc.). La metodologia Waterfall è stata uno dei primi processi di sviluppo del sistema definiti. Un'immagine della metodologia Waterfall è mostrata nella Figura 1.

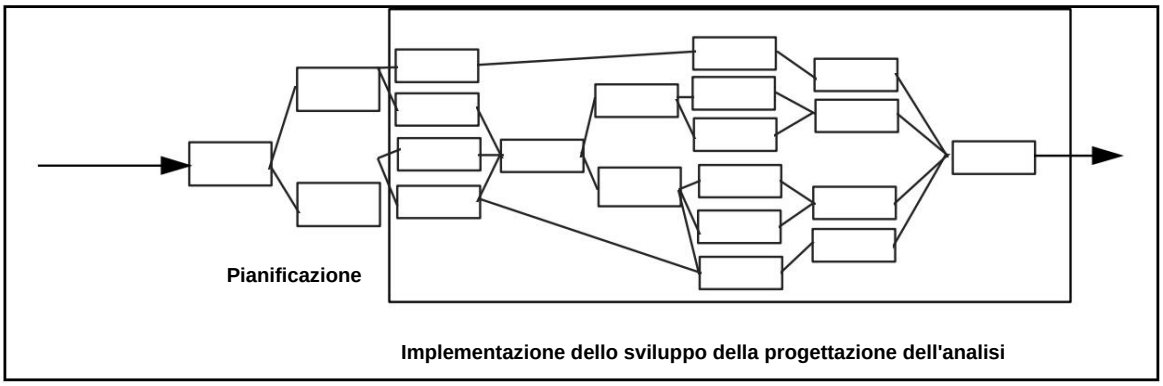

Figura 1: Metodologia a cascata

Sebbene l'approccio a cascata imponga l'uso di processi non definiti, la sua natura lineare è stata il problema più grande. Il processo non definisce come rispondere all'output imprevisto di uno qualsiasi dei processi intermedi.

Barry Boehm12 ha introdotto una metodologia a spirale per affrontare questo problema. Ciascuna delle fasi della cascata si conclude con un'attività di valutazione del rischio e di prototipazione. La metodologia Spiral è mostrata nella Figura 2.

La metodologia Spiral "sbuccia la cipolla", procedendo attraverso "strati" del processo di sviluppo. Un prototipo consente agli utenti di determinare se il progetto è sulla buona strada, dovrebbe essere rimandato alle fasi precedenti o dovrebbe essere terminato. Tuttavia, le fasi e i processi di fase sono ancora lineari. Il lavoro sui requisiti viene ancora eseguito nella fase dei requisiti, il lavoro di progettazione nella fase di progettazione e così via, con ciascuna fase costituita da processi lineari ed esplicitamente definiti.

[12] Boehm, BW 1985. "Un modello a spirale di sviluppo e miglioramento del software", *da Atti di un workshop internazionale sui processi software e sugli ambienti software,* Coto de Caza, Trabuco Canyon, California, 27-29 marzo 1985.

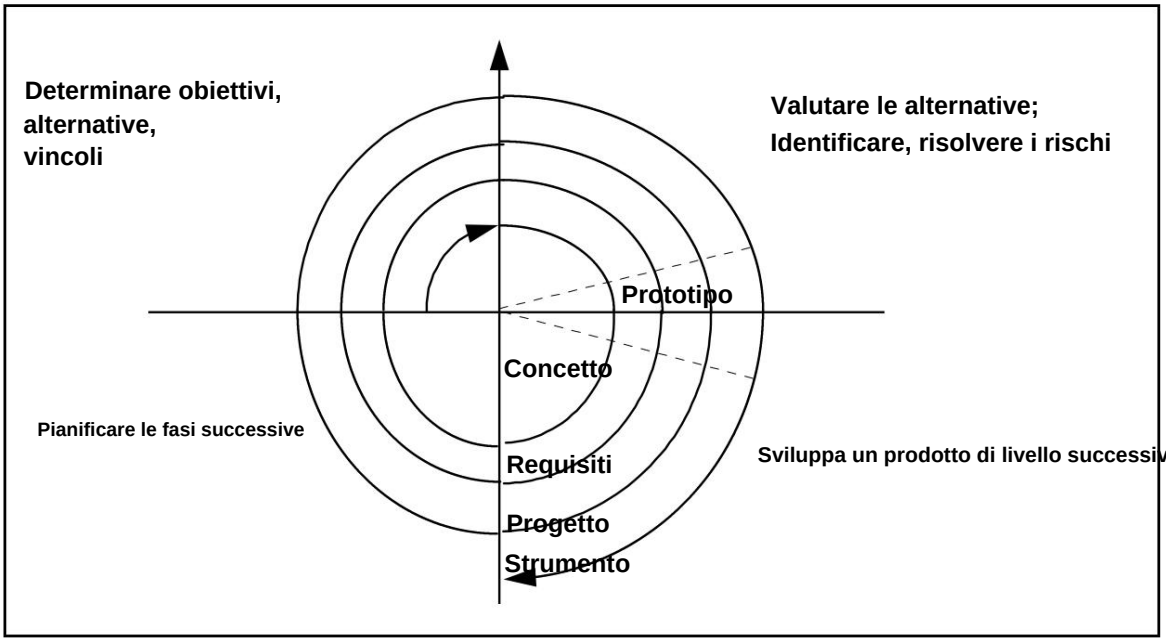

Figura 2: Metodologia a spirale

La metodologia Iterativa migliora la metodologia Spiral. Ogni iterazione è composta da tutte le fasi standard di Waterfall, ma ogni iterazione indirizza solo un set di funzionalità analizzate. I risultati complessivi del progetto sono stati suddivisi in sottosistemi prioritari, ciascuno con interfacce pulite. Utilizzando questo approccio, è possibile testare la fattibilità di un sottosistema e di una tecnologia nelle iterazioni iniziali. Ulteriori iterazioni possono aggiungere risorse al progetto accelerando al tempo stesso la velocità di consegna. Questo approccio migliora il controllo dei costi, garantisce la fornitura di sistemi (anche se sottosistemi) e migliora la flessibilità complessiva. Tuttavia, l'approccio iterativo prevede ancora che i processi di sviluppo sottostanti siano definiti e lineari. Vedere la Figura 3.

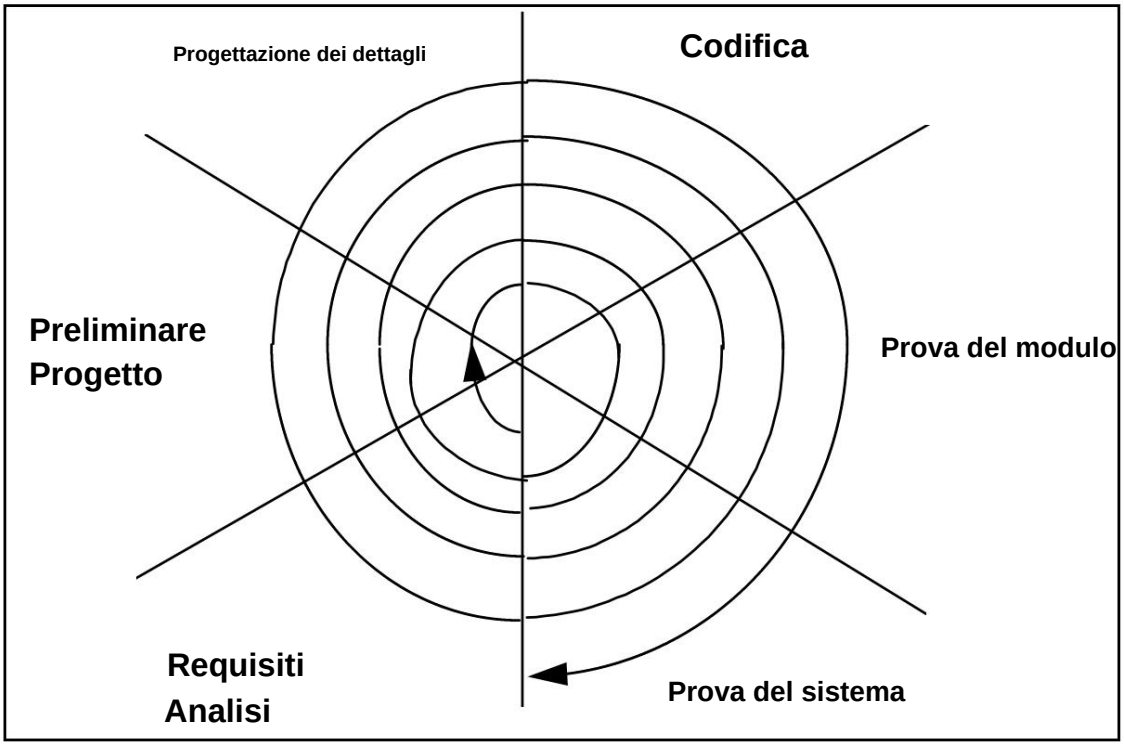

Figura 3: Metodologia iterativa

Dato il contesto complesso e la crescente dipendenza da nuovi sistemi “all’avanguardia”, il rischio sopportato dai progetti di sviluppo del sistema è aumentato e la ricerca di meccanismi per gestire questo rischio si è intensificata.

Si può sostenere che le metodologie attuali sono meglio di niente. Ognuno migliora l'altro. Gli approcci Spirale e Iterativo impiantano meccanismi formali di controllo del rischio per gestire risultati imprevedibili. Viene fornito un quadro per lo sviluppo.

Tuttavia, entrambi si basano sull’errore secondo cui i processi di sviluppo sono processi definiti e prevedibili. Ma nel corso dei progetti si verificano risultati imprevedibili. Il rigore implicito nei processi di sviluppo soffoca la flessibilità necessaria per far fronte ai risultati imprevedibili e rispondere a un ambiente complesso.

Nonostante la loro diffusa presenza nella comunità di sviluppo, la nostra esperienza nel settore mostra che le persone non utilizzano le metodologie se non come mappa di macro processi o per descrizioni dettagliate dei metodi.

Il grafico seguente mostra l'attuale ambiente di sviluppo, utilizzando uno qualsiasi dei processi Waterfall, Spiral o Iterativo. Poiché la complessità delle variabili aumenta anche a un livello moderato, la probabilità di un progetto “di successo” diminuisce rapidamente (un progetto di successo è definito come un sistema che è utile una volta consegnato). Vedere la Figura 4.

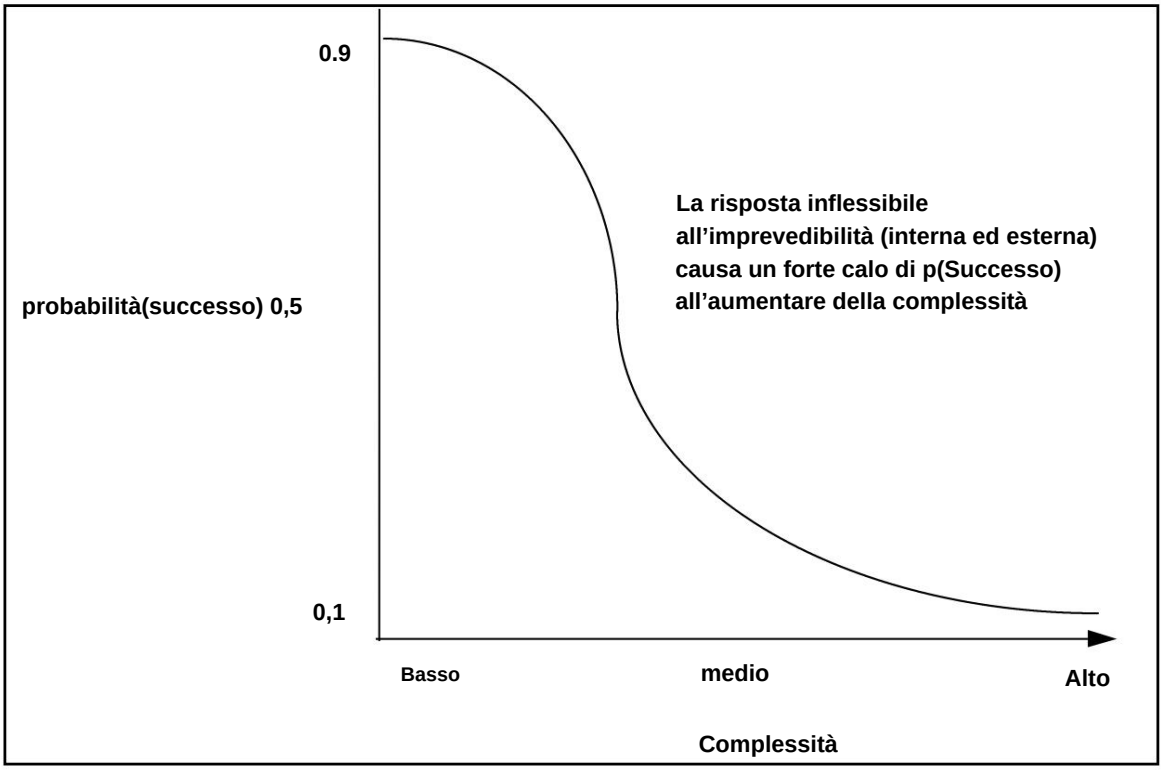

Figura 4
Grafico del rischio/complessità del processo definito

4. Metodologia SCRUM

Il processo di sviluppo del sistema è complicato e complesso. Pertanto sono necessari la massima flessibilità e un controllo adeguato. L'evoluzione favorisce coloro che operano con la massima esposizione ai cambiamenti ambientali e hanno ottimizzato un adattamento flessibile al cambiamento. L'evoluzione deseleziona coloro che si sono isolati dai cambiamenti ambientali e hanno ridotto al minimo il caos e la complessità del loro ambiente.

È necessario un approccio che consenta ai team di sviluppo di operare in modo adattivo all'interno di un ambiente complesso utilizzando processi imprecisi. Lo sviluppo di sistemi complessi avviene in circostanze in rapido cambiamento. Produrre sistemi ordinati in circostanze caotiche richiede la massima flessibilità. Più il team di sviluppo si avvicina al limite del caos, pur mantenendo l'ordine, più competitivo e utile sarà il sistema risultante. Langton ha modellato questo effetto in simulazioni al computer13 e il suo lavoro lo ha fornito come teorema fondamentale nella teoria della complessità.

[13] Langton, Christopher. Vita artificiale. In Artificial Life, Volume VI: SFI Studies in the Sciences of Complexity (Ed. C. Langton) Addison-Wesley, 1988.

La metodologia potrebbe essere il fattore più importante nel determinare la probabilità di successo. Le metodologie che incoraggiano e supportano la flessibilità hanno un alto grado di tolleranza per i cambiamenti in altre variabili. Con queste metodologie, il processo di sviluppo è considerato imprevedibile all'inizio e vengono messi in atto meccanismi di controllo per gestire l'imprevedibilità.

Se rappresentiamo graficamente la relazione tra complessità ambientale e probabilità di successo con una metodologia flessibile che incorpori controlli e gestione del rischio, la tolleranza al cambiamento è più duratura. Vedere la Figura 5.

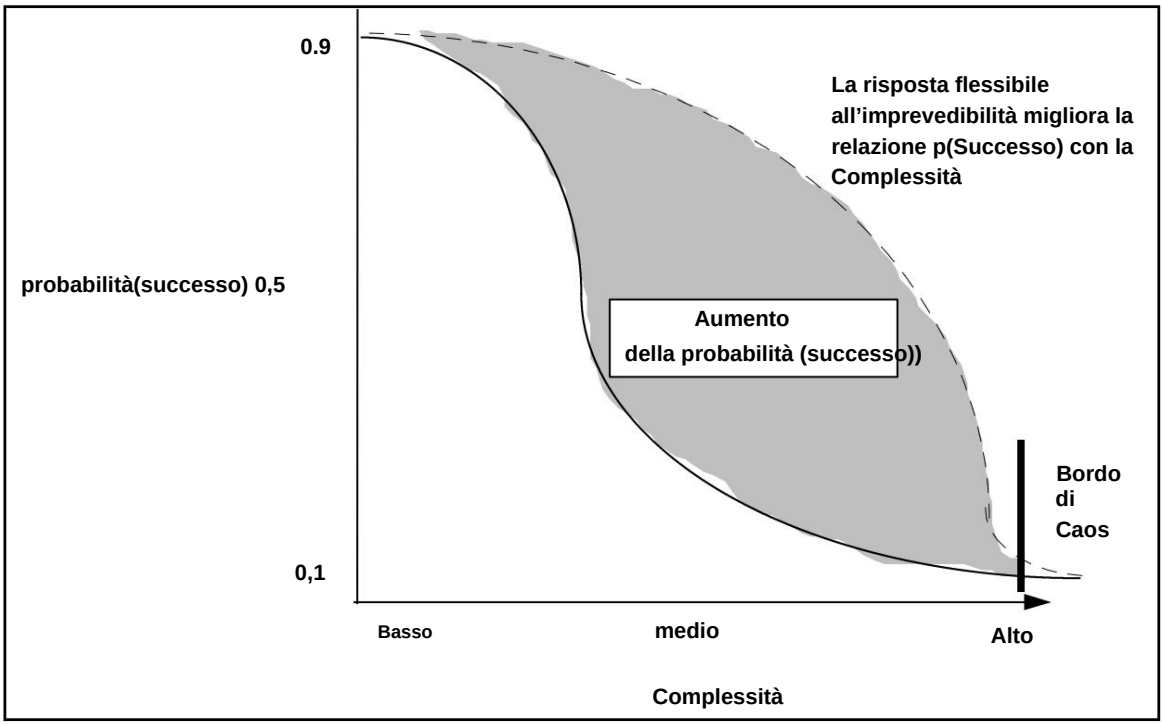

Figura 5 – Grafico di confronto rischio/complessità

Le figure 4 e 5 riflettono le esperienze di sviluppo software presso ADM, Easel, VMARK, Borland e praticamente ogni altro sviluppatore di software "confezionato". Queste organizzazioni hanno abbracciato il rischio e la complessità ambientale durante i progetti di sviluppo.
Sono stati riscontrati un maggiore impatto del prodotto, progetti di successo e incrementi di produttività. Viene creato il miglior software possibile.

Le metodologie Waterfall e Spiral stabiliscono il contesto e la definizione dei risultati finali all'inizio di un progetto. Le metodologie SCRUM e iterative pianificano inizialmente il contesto e la definizione ampia del deliverable, quindi evolvono il deliverable durante il progetto in base all'ambiente. SCRUM riconosce che i processi di sviluppo sottostanti sono definiti in modo incompleto e utilizza meccanismi di controllo per migliorare la flessibilità.

La differenza principale tra l'approccio definito (a cascata, a spirale e iterativo) e quello empirico (SCRUM) è che l'approccio SCRUM presuppone che i processi di analisi, progettazione e sviluppo nella fase Sprint siano imprevedibili. Un meccanismo di controllo viene utilizzato per gestire l'imprevedibilità e controllare il rischio. I risultati sono flessibilità, reattività e affidabilità. Vedere la Figura 6.

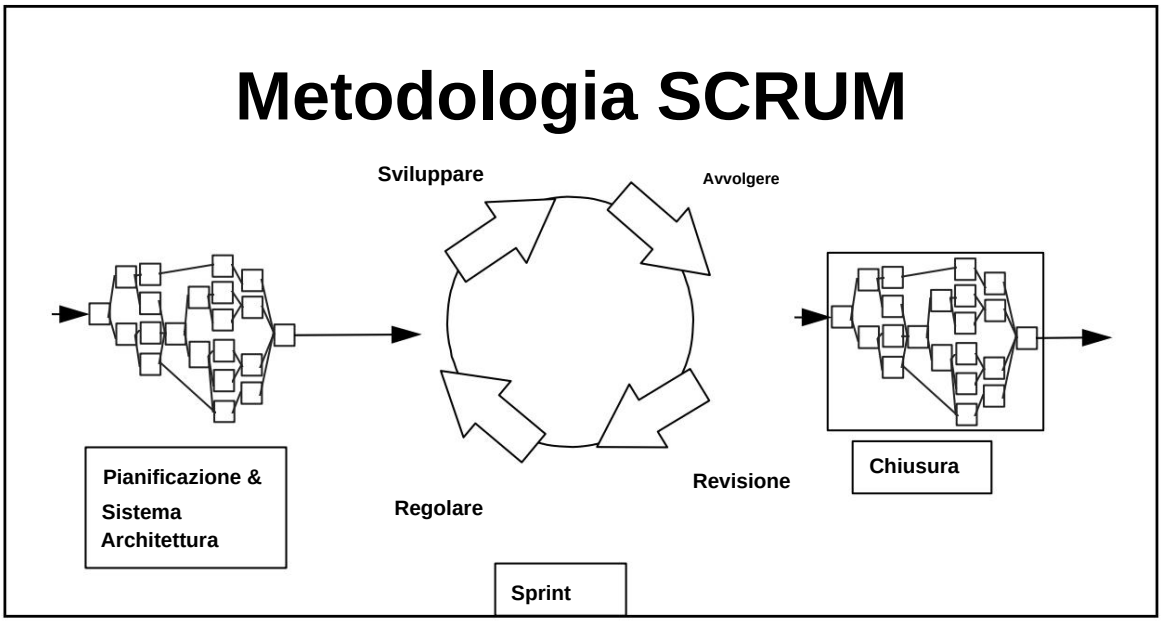

Figura 6: Metodologia SCRUM

Caratteristiche della metodologia SCRUM sono:

La prima e l'ultima fase (Pianificazione e Chiusura) sono costituite da processi definiti, in cui tutti i processi, input e output sono ben definiti. La conoscenza di come eseguire questi processi è esplicita. Il flusso è lineare, con alcune iterazioni in fase di pianificazione.

La fase Sprint è un processo empirico. Molti dei processi nella fase di sprint non sono identificati o non controllati. Viene trattato come una scatola nera che richiede controlli esterni. Di conseguenza, i controlli, inclusa la gestione del rischio, vengono posti in ogni iterazione della fase Sprint per evitare il caos massimizzando al tempo stesso la flessibilità.

Gli sprint sono non lineari e flessibili. Ove disponibile, viene utilizzata la conoscenza esplicita del processo; altrimenti la conoscenza tacita, la prova e l'errore vengono utilizzati per costruire la conoscenza del processo. Gli sprint vengono utilizzati per evolvere il prodotto finale.

Il progetto è aperto all'ambiente fino alla fase di chiusura. Il deliverable può essere modificato in qualsiasi momento durante le fasi di Planning e Sprint del progetto. Il progetto rimane aperto alla complessità ambientale, comprese le pressioni competitive, temporali, qualitative e finanziarie, durante queste fasi.

Il risultato finale viene determinato durante il progetto in base all'ambiente.

La Figura 7 confronta le caratteristiche primarie di SCRUM con quelle di altre metodologie.

	Cascata	Spirale	Iterativo	MISCHIA
Definito processi	Necessario	Necessario	Necessario	Pianificazione & Solo chiusura
Prodotto finale	Determinato durante la pianificazione	Determinato durante la pianificazione	Impostato durante il progetto	Impostato durante il progetto
Costo del progetto	Determinato durante la pianificazione	Parzialmente variabile	Impostato durante il progetto	Impostato durante il progetto
Data di completamento	Determinato durante la pianificazione	Parzialmente variabile	Impostato durante il progetto	Impostato durante il progetto
Reattività all'ambiente	Solo pianificazione	Pianificazione innanzitutto	Alla fine di ogni iterazione	**Per tutto**
Flessibilità del team, creatività	Limitato: approccio da libro di cucina	Limitato: approccio da libro di cucina	Limitato: approccio da libro di cucina	**Illimitato durante le iterazioni**
Trasferimento di conoscenza	Formazione prima del progetto	Formazione prima del progetto	Formazione prima del progetto	**Lavoro di squadra durante il progetto**
Probabilità di successo	Basso	Medio basso	medio	**Alto**

Figura 7: Confronto metodologico

4.1 Fasi della ~~MISCHIA~~ SCRUM

SCRUM ha i seguenti gruppi di fasi:

4.1.1. Pianificazione

pre-partita : definizione di una nuova versione in base al backlog attualmente noto, insieme a una stima del suo programma e dei costi. Se si sta sviluppando un nuovo sistema, questa fase consiste sia nella concettualizzazione che nell'analisi. Se un sistema esistente viene potenziato, questa fase consiste in un'analisi limitata.

Architettura: progettare come verranno implementati gli elementi del backlog. Questa fase include la modifica dell'architettura del sistema e la progettazione di alto livello.

4.1.2. Sprint

di sviluppo *del gioco :* sviluppo di nuove funzionalità di rilascio, con un costante rispetto delle variabili di tempo, requisiti, qualità, costo e concorrenza.

L'interazione con queste variabili definisce la fine di questa fase. Esistono molteplici sprint o cicli di sviluppo iterativi utilizzati per far evolvere il sistema.

4.1.3. Chiusura

post-partita : preparazione al rilascio, inclusa la documentazione finale, test graduali pre-rilascio e rilascio.

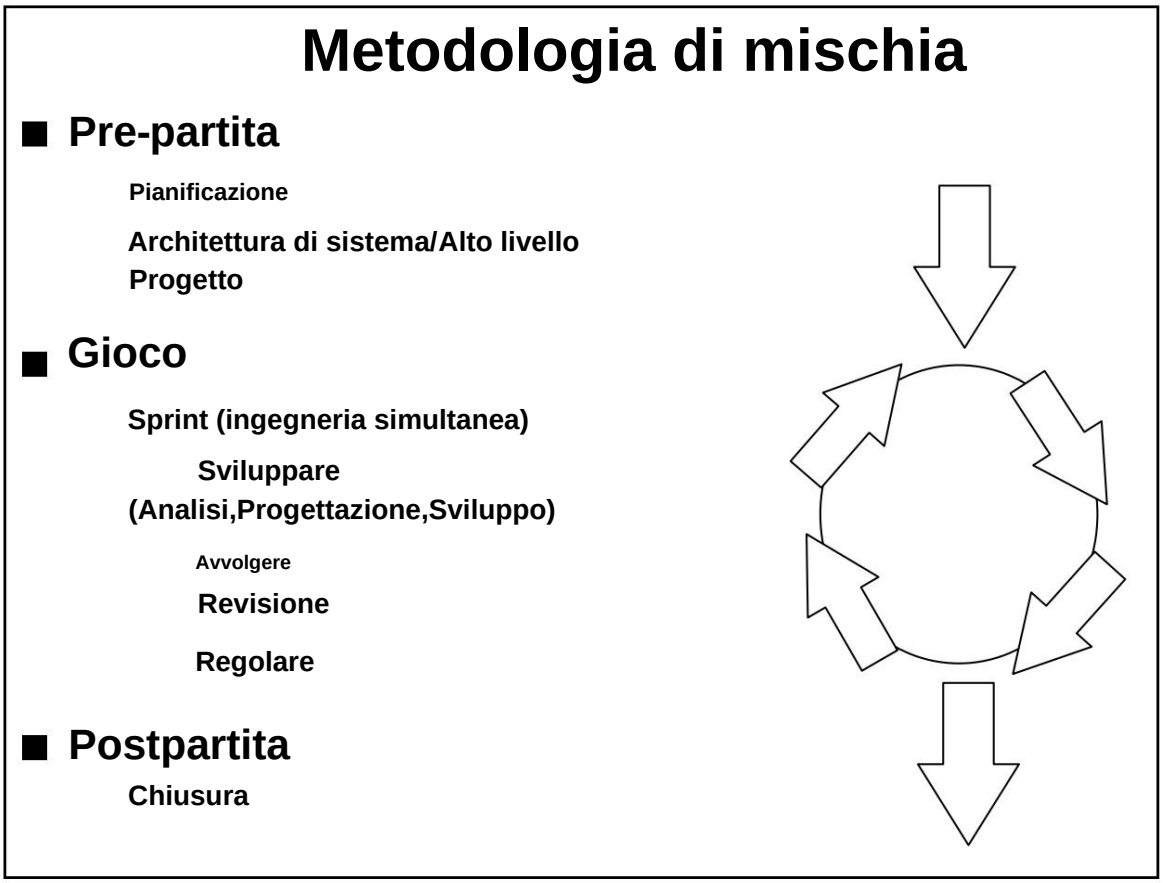

Figura 9

4.2 Fasi della fase

Ciascuna fase prevede i seguenti passaggi:

4.2.1. Pianificazione

dello sviluppo di un elenco completo DEL BACKLOG ~~degli arretrati.~~

Definizione della data di consegna e funzionalità di uno o più rilasci.

Selezione della release più opportuna per lo sviluppo immediato.

Mappatura dei pacchetti di prodotti (oggetti) per gli elementi del backlog nella versione selezionata.

Definizione dei team di progetto per la realizzazione della nuova versione.

Valutazione del rischio e adeguati controlli del rischio.

Revisione ed eventuale aggiustamento delle voci e dei pacchetti arretrati.

Convalida o riselezione degli strumenti e dell'infrastruttura di sviluppo.

Stima dei costi di rilascio, inclusi sviluppo, materiale collaterale, marketing, formazione e implementazione.

Verifica dell'approvazione della direzione e dei finanziamenti.

4.2.2. Architettura/ Progettazione di alto livello

Revisione degli elementi del backlog assegnati.

Identificare le modifiche necessarie per implementare gli elementi del backlog.

Eseguire l'analisi del dominio nella misura necessaria per creare, migliorare o aggiornare i modelli di dominio per riflettere il nuovo contesto e i requisiti del sistema.

Perfezionare l'architettura del sistema per supportare il nuovo contesto e i nuovi requisiti.

Identificare eventuali problemi o problemi nello sviluppo o nell'implementazione delle modifiche

Riunione di revisione del progetto, ogni team presenta l'approccio e le modifiche per implementare ciascun elemento del backlog. Riassegnare le modifiche come richiesto.

4.2.3. Sviluppo (Sprint)

La fase di sviluppo è un ciclo iterativo del lavoro di sviluppo. Il management determina che il tempo, la concorrenza, la qualità o la funzionalità siano rispettati, le iterazioni siano completate e si verifichi la fase di chiusura. Questo approccio è noto anche come Ingegneria Concorrente. Lo sviluppo si compone dei seguenti macro processi:

Incontro con i team per rivedere i piani di rilascio.

Distribuzione, revisione e adeguamento degli standard ai quali il prodotto sarà conforme.

Sprint iterativi, finché il prodotto non è ritenuto pronto per la distribuzione.

Uno Sprint è un insieme di attività di sviluppo condotte in un periodo predefinito, solitamente da una a quattro settimane. L'intervallo si basa sulla complessità del prodotto, sulla valutazione del rischio e sul grado di supervisione desiderato. La velocità e l'intensità dello Sprint sono determinate dalla durata selezionata dello Sprint. Il rischio viene valutato continuamente e vengono messi in atto adeguati controlli e risposte al rischio. Ogni Sprint è composto da uno o più team che eseguono le seguenti operazioni:

Sviluppo: definizione delle modifiche necessarie per l'implementazione dei requisiti del backlog nei pacchetti, apertura dei pacchetti, esecuzione dell'analisi del dominio, progettazione, sviluppo, implementazione, test e documentazione delle modifiche. Lo sviluppo consiste nel micro processo di scoperta, invenzione e implementazione.
Wrap: chiusura dei pacchetti, creazione di una versione eseguibile delle modifiche e modalità di implementazione dei requisiti del backlog.
Revisione: tutti i team si incontrano per presentare il lavoro e rivedere i progressi, sollevando e risolvendo questioni e problemi, aggiungendo nuovi elementi di arretrato. Il rischio viene esaminato e vengono definite le risposte appropriate.
Modifica: consolidamento delle informazioni raccolte dalla riunione di revisione nei pacchetti interessati, inclusi aspetto diverso e nuove proprietà.

Ogni Sprint è seguito da una revisione, le cui caratteristiche sono:

- L'intero team e la direzione del prodotto sono presenti e partecipano.
- La revisione può includere clienti, vendite, marketing e altri.
- La revisione riguarda i sistemi funzionali ed eseguibili che comprendono gli oggetti assegnati a quel team e includono le modifiche apportate per implementare gli elementi del backlog.
- Il modo in cui gli elementi del backlog vengono implementati dalle modifiche può essere modificato in base alla revisione.
- Nuovi elementi del backlog possono essere introdotti e assegnati ai team come parte della revisione, modificando il contenuto e la direzione dei risultati finali.
- Il momento della revisione successiva viene determinato in base al progresso e alla complessità. Gli Sprint hanno solitamente una durata da 1 a 4 settimane.

4.2.4. Chiusura

Quando il team di gestione ritiene che le variabili tempo, competizione, requisiti, costi e qualità concorrono affinché avvenga un nuovo rilascio, dichiara il rilascio “chiuso” ed entra in questa fase. Questa fase prepara il prodotto sviluppato per il rilascio generale.
L'integrazione, il test del sistema, la documentazione per l'utente, la preparazione del materiale di formazione e la preparazione del materiale di marketing sono tra le attività di chiusura.

Scrum

4.3. Controlli ~~MISCHIA~~

Operare ai margini del caos (imprevedibilità e complessità) richiede controlli di gestione per evitare di cadere nel caos. La metodologia SCRUM incorpora questi controlli generali e sciolti, utilizzando tecniche OO per la costruzione effettiva dei risultati finali.

Il rischio è il controllo primario. La valutazione del rischio porta a cambiamenti in altri controlli e risposte da parte del team.

I controlli nella metodologia SCRUM sono:

BACKLOG
Arretrato: requisiti di funzionalità del prodotto che non sono adeguatamente affrontati dalla versione corrente del prodotto. Bug, difetti, miglioramenti richiesti dai clienti, funzionalità di prodotti competitivi, funzionalità di vantaggio competitivo e aggiornamenti tecnologici sono elementi arretrati.
BACKLOG ITEMS

Rilascio/miglioramento: elementi del backlog che in un determinato momento rappresentano un rilascio fattibile in base alle variabili di requisiti, tempo, qualità e concorrenza.

Pacchetti: componenti o oggetti del prodotto che devono essere modificati per implementare un elemento del backlog in una nuova versione.

Modifiche: modifiche che devono essere apportate a un pacchetto per implementare un elemento del backlog.

Problemi: problemi tecnici che si verificano e devono essere risolti per implementare una modifica.

Rischi: i rischi che influiscono sul successo del progetto vengono continuamente valutati e le risposte pianificate. Altri controlli sono influenzati a seguito della valutazione del rischio.

Soluzioni: soluzioni ai problemi e ai rischi, che spesso portano a cambiamenti.

Problemi: progetto generale e problemi del progetto che non sono definiti in termini di pacchetti, modifiche e problemi.

Questi controlli vengono utilizzati nelle varie fasi di SCRUM. IL MANAGEMENT La direzione utilizza questi controlli per gestire il backlog. I team utilizzano questi controlli per gestire cambiamenti e problemi.
Sia il management che i team gestiscono congiuntamente problemi, rischi e soluzioni. Questi controlli vengono rivisti, modificati e riconciliati ad ogni riunione di revisione dello Sprint.

4.4 Deliverable SCRUM

Il prodotto consegnato è flessibile. Il suo DEL DELIVERABLE contenuto è determinato da variabili ambientali, tra cui tempo, concorrenza, costo o funzionalità. I fattori determinanti sono l'intelligenza del mercato, il contatto con i clienti e l'abilità degli sviluppatori. Durante il progetto si verificano frequenti aggiustamenti ai contenuti consegnabili in risposta all'ambiente. Il risultato finale può essere determinato in qualsiasi momento durante il progetto.

4.5 Team di progetto SCRUM

Il team che lavora alla nuova versione comprende sviluppatori a tempo pieno e soggetti esterni che saranno interessati dalla nuova versione, ad esempio marketing, vendite e clienti. Nei processi di rilascio tradizionali, questi ultimi gruppi vengono tenuti lontani dai team di sviluppo per paura di complicare eccessivamente il processo e fornire interferenze "non necessarie". L'approccio SCRUM, tuttavia, accoglie e facilita il loro coinvolgimento controllato a intervalli prestabiliti, poiché ciò aumenta la probabilità che il contenuto e la tempistica del rilascio siano appropriati, utili e commerciabili.

Per ogni nuova versione vengono formati i seguenti team:

Gestione: guidato dal Product Manager, definisce i contenuti iniziali e le tempistiche del rilascio, quindi ne gestisce l'evoluzione man mano che il progetto avanza e le variabili cambiano.
~~La gestione~~ si occupa del backlog, dei rischi e dei contenuti rilasciati.
IL MANAGEMENT

Team di sviluppo: i team di sviluppo sono piccoli e ciascuno contiene sviluppatori, documentatori e personale di controllo qualità. Vengono utilizzate una o più squadre composte da tre a sei persone ciascuna. A ciascuno viene assegnato un insieme di pacchetti (o oggetti), inclusi tutti gli elementi del backlog relativi a ciascun pacchetto. Il team definisce le modifiche necessarie per implementare l'elemento del backlog nei pacchetti e gestisce tutti i problemi relativi alle modifiche. I team possono essere derivati dal punto di vista funzionale (a cui vengono assegnati i pacchetti che indirizzano set specifici di funzionalità del prodotto) o derivati dal sistema (a cui vengono assegnati livelli univoci del sistema). I membri di ciascun team vengono selezionati in base alla loro conoscenza ed esperienza riguardo a set di pacchetti o competenza nel dominio.

4.6 Caratteristiche ~~della MISCHIA~~ DI SCRUM

La metodologia SCRUM è una metafora del gioco del Rugby. Il rugby si è evoluto dal calcio inglese (calcio) sotto l'intensa pressione del gioco:

> Lo studente di rugby William Webb Ellis, 17 anni, inaugura un nuovo gioco le cui regole saranno codificate nel 1839. Giocando a calcio per il college di 256 anni nell'East
> Warwickshire, Ellis vede che il tempo sta scadendo *con la sua squadra alle spalle, quindi raccoglie la palla e corre con essa sfidando le regole.*
> **La cronologia popolare,** Henry Holt and Company, Inc. Copyright © 1992.

I progetti SCRUM hanno le seguenti caratteristiche:

Deliverable flessibile: il contenuto del deliverable è dettato dall'ambiente.

Programma flessibile: il risultato finale potrebbe essere richiesto prima o dopo rispetto a quanto inizialmente previsto.

Piccole squadre: ogni squadra non ha più di 6 membri. Potrebbero esserci più team all'interno di un progetto.

Revisioni frequenti: i progressi del team vengono rivisti con la frequenza richiesta dalla complessità ambientale e dal rischio (di solito cicli da 1 a 4 settimane). Un eseguibile funzionale deve essere preparato da ciascun team per ogni revisione. (comica)

Collaborazione: durante il progetto è prevista la collaborazione intra e inter.

Object Oriented: ogni squadra affronterà una serie di oggetti correlati, con interfacce e comportamenti chiari.

La metodologia SCRUM condivide molte caratteristiche con lo sport del Rugby:

Il contesto è stabilito dal campo di gioco (ambiente) e dalle regole del rugby (controlli).

Il ciclo primario ~~sta spostando~~ MUOVE la palla in avanti.

Il rugby si è evoluto infrangendo le regole del calcio e adattandosi all'ambiente.

Il gioco non finisce finché l'ambiente non lo impone (necessità aziendale, concorrenza, funzionalità, calendario).

5. Vantaggi della Metodologia SCRUM

Le metodologie di sviluppo tradizionali sono progettate solo per rispondere all'imprevedibilità degli ambienti esterni e di sviluppo all'inizio di un ciclo di miglioramento. Gli approcci più recenti, come la metodologia a spirale di Boehm e le sue varianti, sono ancora limitati nella loro capacità di rispondere alle mutevoli esigenze una volta avviato il progetto.

La metodologia SCRUM, d'altra parte, è progettata per essere abbastanza flessibile in ogni sua parte. Fornisce meccanismi di controllo per pianificare il rilascio di un prodotto e quindi gestire le variabili man mano che il progetto avanza. Ciò consente alle organizzazioni di modificare il progetto e i risultati finali in qualsiasi momento, fornendo la versione più appropriata.

La metodologia SCRUM lascia liberi gli sviluppatori di ideare le soluzioni più ingegnose durante tutto il progetto, man mano che avviene l'apprendimento e l'ambiente cambia.

Piccoli team collaborativi di sviluppatori sono in grado di condividere la conoscenza tacita sui processi di sviluppo. Viene fornito un ambiente di formazione eccellente per tutte le parti.

La tecnologia Object Oriented fornisce la base per la metodologia SCRUM. Gli oggetti, o funzionalità del prodotto, offrono un ambiente discreto e gestibile. Il codice procedurale, con le sue numerose e intrecciate interfacce, è inappropriato per la metodologia SCRUM. SCRUM può essere applicato selettivamente a sistemi procedurali con interfacce pulite e un forte orientamento ai dati.

6. Stima del progetto SCRUM

I progetti SCRUM possono essere stimati utilizzando la stima dei punti funzione standard. Tuttavia, è consigliabile stimare la produttività a circa il doppio del parametro attuale. La stima è tuttavia solo iniziale, poiché il calendario complessivo e i costi vengono determinati dinamicamente in risposta ai fattori ambientali.

Le nostre osservazioni ci hanno portato a concludere che i progetti SCRUM hanno sia velocità che accelerazione. In termini di funzioni consegnate o di elementi in arretrato completati:

1. la velocità e l'accelerazione iniziali sono basse man mano che l'infrastruttura viene costruita/
2. modificata man mano che la funzionalità di base viene inserita negli oggetti,
3. l'accelerazione aumenta l'accelerazione diminuisce e la velocità rimane elevata in modo sostenibile

È necessario un ulteriore sviluppo delle metriche per i processi empirici.

7. Metodologie di sviluppo del sistema: definite o empiriche

Lo sviluppo del sistema è l'atto di creare un costrutto logico implementato come logica e dati sui computer. Il costrutto logico è costituito da input, processi e output, sia macro (intero costrutto) che micro (passaggi intermedi all'interno dell'intero costrutto). Il tutto è noto come sistema implementato.

Molti artefatti vengono creati durante la creazione del sistema. Gli artefatti possono essere utilizzati per guidare il pensiero, verificare la completezza e creare una traccia di controllo. Gli artefatti sono costituiti da documenti, modelli, programmi, casi di test e altri risultati finali creati prima della creazione del sistema implementato. Quando disponibile, un *metamodello* definisce il contenuto semantico degli artefatti del modello. *La notazione* descrive le convenzioni grafiche e di documentazione utilizzate per costruire i modelli.

L'approccio utilizzato per sviluppare un sistema è noto come *metodo.* Un *metodo* descrive le attività coinvolte nella definizione, costruzione e implementazione di un sistema; un *metodo* è una struttura. Poiché un *metodo* è un processo logico per costruire sistemi (processo), è noto come metaprocesso *(un* processo per modellare i processi).

Un *metodo* ha componenti micro e macro. Le macro componenti definiscono il flusso generale e la struttura sequenziata nel tempo per l'esecuzione del lavoro. I microcomponenti includono *regole generali di progettazione, modelli* e *regole pratiche. Le regole generali di progettazione* stabiliscono le proprietà da raggiungere o da evitare nella progettazione o gli approcci generali da adottare durante la costruzione di un sistema. *I pattern* sono soluzioni che possono essere applicate a un tipo di attività di sviluppo; sono soluzioni in attesa di problemi che si verificano durante un'attività in un metodo. *Le regole pratiche* consistono in un insieme generale di suggerimenti e suggerimenti.

La Figura 1 visualizza la relazione tra il Metodo, gli Artefatti e il Sistema.

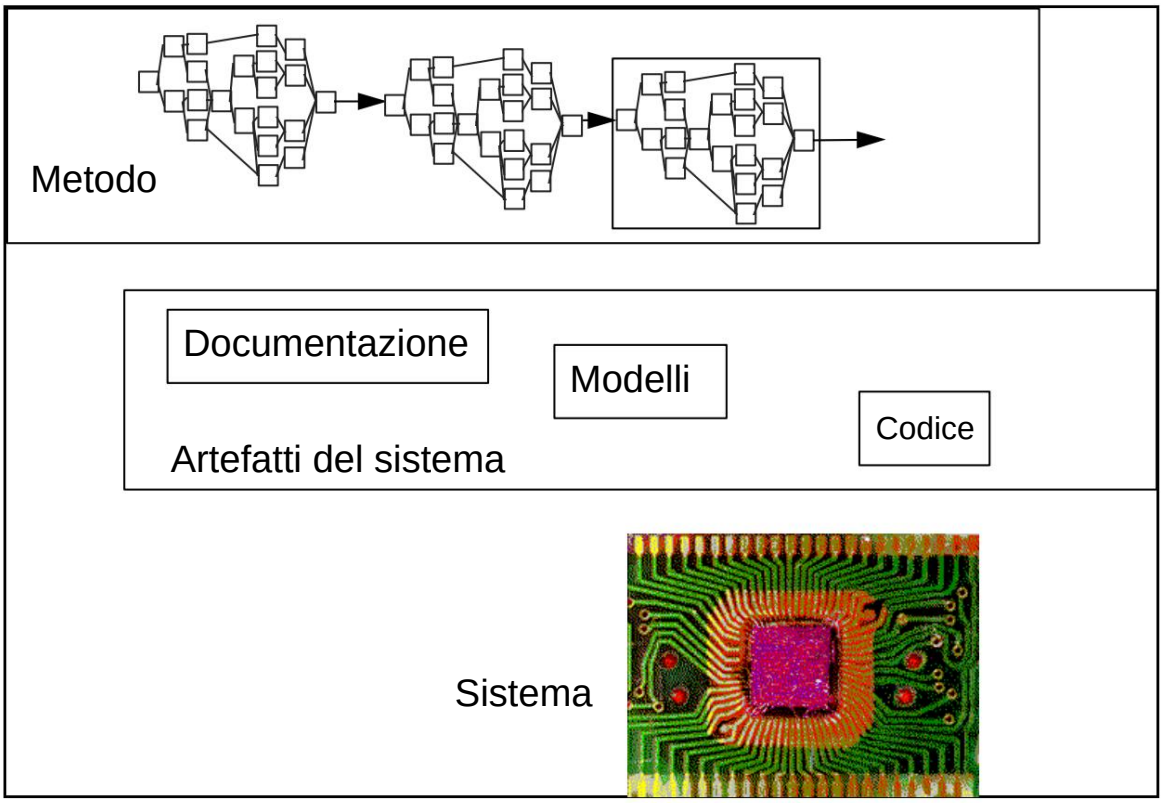

Figura 1

Applicando concetti dal controllo dei processi industriali al campo dello sviluppo di sistemi, *i metodi* possono essere classificati come “teorici” (completamente definiti) o “empirici” (scatola nera).

Classificare correttamente i metodi di sviluppo dei sistemi è fondamentale. La struttura appropriata di un *metodo* per costruire un particolare tipo di sistema dipende dal fatto che il metodo sia teorico o empirico.

I modelli dei *processi teorici* derivano dai *principi primi,* utilizzando gli equilibri materiali ed energetici e le leggi fondamentali per determinare il modello. Affinché un *metodo* di sviluppo dei sistemi possa essere classificato come teorico, deve essere conforme a questa definizione.

Vengono derivati modelli di ***processi empirici*** classificando gli input e gli output osservati e definendo i controlli che ne fanno sì che si verifichino entro limiti prescritti. La modellazione empirica del processo implica la costruzione di un modello di processo rigorosamente a partire da dati di input/output ottenuti sperimentalmente, senza ricorrere ad alcuna legge riguardante la natura fondamentale e le proprietà del sistema. Non è necessaria *alcuna conoscenza a priori* del processo (anche se può essere utile); un sistema viene trattato come una scatola nera.

Le caratteristiche principali sia della modellazione teorica che empirica sono dettagliate nella Tabella 1.

Modellazione teorica 1. Di	**La modellazione empirica**
solito comporta meno misurazioni; richiede sperimentazione solo per la stima di parametri del modello sconosciuti.	richiede misurazioni approfondite, perché si basa interamente sulla sperimentazione per lo sviluppo del modello.
2. Fornisce informazioni sullo stato interno del processo.	Fornisce informazioni solo su quella parte del processo che può essere influenzata dall'azione di controllo.
3. Promuove la comprensione fondamentale del funzionamento interno del processo.	Tratta il processo come una “scatola nera”.
4. Richiede una conoscenza del processo abbastanza accurata e completa.	Non richiede una conoscenza così dettagliata; solo che i dati di output siano ottenibili in risposta ai cambiamenti di input.
5. Non particolarmente utile per processi poco compresi e/o complessi.	Molto spesso si rivela l'unica alternativa per modellare il comportamento di processi poco compresi e/o complessi.
6. Produce naturalmente modelli di processo sia lineari che non lineari.	Richiede metodi speciali per produrre modelli non lineari.

Tabella 1

Dopo l'ispezione, affermiamo che il processo di sviluppo dei sistemi è empirico:

1. Non sono presenti principi primi applicabili 2. Il processo sta appena iniziando a essere compreso 3. Il processo è complesso 4. Il processo sta cambiando

La maggior parte dei metodologi concorda con questa affermazione; "...non puoi aspettarti che un *metodo* ti dica tutto quello che devi fare. Scrivere software è un processo creativo, come la pittura, la scrittura o l'architettura... ... (un *metodo)* fornisce una struttura che spiega come procedere e identifica i luoghi in cui è necessaria la creatività. Ma devi ancora fornire la creatività…."[14]

Classificare i metodi di sviluppo dei sistemi come empirici è fondamentale per la gestione efficace del processo di sviluppo dei sistemi.

Se *i metodi* di sviluppo dei sistemi vengono classificati come *empirici,* misurazioni e controlli sono necessari perché è chiaro che il funzionamento interno del *metodo* è così vagamente definito che non si può contare sul fatto che operino in modo prevedibile.

In passato, *i metodi* venivano forniti e applicati come se fossero teorici. Di conseguenza, non si è fatto affidamento sulle misurazioni e non sono stati utilizzati i controlli dipendenti dalle misurazioni.

Molti dei problemi nello sviluppo dei sistemi si sono verificati a causa di questa categorizzazione errata. Quando un processo scatola nera viene trattato come un processo completamente definito, si verificano risultati imprevedibili. Inoltre, non esistono controlli per misurare e rispondere all'imprevedibilità.

In pratica, in più aziende su più progetti all'interno di alcune aziende, è stato osservato che SCRUM fornisce una soluzione praticabile a questi problemi. I progetti vengono consegnati in tempo e spesso superano le aspettative sia degli utenti che del management.
Sebbene lavorare in un team di sviluppo SCRUM sia intenso, gli sviluppatori sono ricompensati da un elevato spirito di squadra, da un profondo senso di realizzazione e dalla sensazione che lo sviluppo possa essere un'esperienza piacevole e soddisfacente.

Riferimenti

1. Gruppo Aberdeen. Aggiornamento alla metodologia ISV per lo sviluppo di applicazioni aziendali. Punto di vista del prodotto 8:17, 7 dicembre 1995.

2. Bach, James. "Evoluzione dei processi in un mondo pazzo." Borland International, Scotts Valle, California.

3. Bach, James. "La sfida del software "abbastanza buono", programmatore americano, Ottobre 1995.

4. Boehm, BW 1985. "A Spiral Model of Software Development and Enhancement", *dagli Atti di un workshop internazionale sui processi software e sugli ambienti software,* Coto de Caza, Trabuco Canyon, California, 27-29 marzo 1985.

[14] James Rumbaugh, ottobre 1995, "Cos'è un metodo"., Object Journal

5. Booch, Grady. Analisi orientata agli oggetti e progettazione con applicazioni. La Benjamin/
Cummings Publishing Company, Inc., 1994, p. 8

6. Booch, Grady. Soluzioni a oggetti: gestire il progetto orientato agli oggetti. Addison-
Wesley, 1995.

7. Brooks, FP "Nessuna soluzione miracolosa: l'essenza e gli incidenti dell'ingegneria del software."
Computer 20:4:10-19, aprile 1987.

8. Coplien, J. "Artigianato del software Borland: un nuovo sguardo a processo, qualità e produttività".
Atti della quinta conferenza annuale Borland International, 5 giugno 1994. Orlando, Florida.

9. DeGrace, P. e Hulet Stahl, L. 1990. *Problemi malvagi, soluzioni giuste.*
Yourdon Press

10. Gartner, Lisa. Il Primer per Principianti. Radcliffe Rugby Football Club, 1996 <http://
vail.al.arizona.edu/rugby/rad/rookie_primer.html>

11. Gleick, J. 1987. *Caos, creazione di una nuova scienza.* Libri dei pinguini.

12.Graham , Ian. Migrazione alla tecnologia a oggetti. Addison-Wesley, 1994.

13. Kahn, D. e Sutherland, J. marzo-aprile 1994. “Object Insider: iniziamo sotto-
promettente e con risultati eccessivi su OT." Rivista di oggetti.

14. Langton, Christopher. Vita artificiale. In Artificial Life, Volume VI: SFI Studies in the Sciences of
Complexity (Ed. C. Langton) Addison-Wesley, 1988.

15. Nonaka, Ikujiro e Takeuchi, Hirotaka. 1995. *The Knowledge Creation Company: come le aziende
giapponesi creano le dinamiche dell'innovazione,* Oxford University Press.

16. Ogunnaike, B. 1994. *Dinamica dei processi, modellazione e controllo.* Università di Oxford
Premere.

17. Pittman, Matteo. Lezioni apprese nella gestione dello sviluppo orientato agli oggetti.
IEEE Software, gennaio 1993, pagine 43-53.

18. Rumbaugh, ottobre 1995, "Cos'è un metodo". Giornale di orientamento agli oggetti
Programmazione.

19. Schwaber, Ken. "Caos controllato: vivere al limite." Programmatore americano,
aprile 1996.

20. Sutherland, Jeff. Home Page di ScrumWeb: una guida al processo di sviluppo di SCRUM. Pagina Web sulla tecnologia degli oggetti di Jeff Sutherland, 1996 <http://www.tiac.net/users/jsuth/scrum/index.html>

21. Takeuchi, Hirotaka e Nonaka, Ikujiro. Gennaio-febbraio 1986. “Il Nuovo Nuovo Gioco di sviluppo prodotto." Revisione aziendale di Harvard.