# Introduzione a Scrum

### Scrum in 100 parole

- È un processo agile che permette di focalizzarci sulla consegna del massimo valore di business nel minor tempo possibile.
- Ci permette di verificare rapidamente e ripetutamente il software funzionante (da ogni 2 settimane a ogni mese).
- Il business definisce le priorità. Il team si auto-organizza per trovare il modo migliore per consegnare le funzionalità a più alto valore.
- Ogni due settimane o un mese chiunque può vedere un vero software funzionante e decider se rilasciarlo agli utenti o continuare a migliorarlo per un altro Sprint.

### Caratteristiche

- Gruppi di lavoro auto-organizzati
- I prodotti sono sviluppati in una progressione di cicli "sprint" da 2-4 settimane
- I requisiti sono definiti in una lista chiamata "Product Backlog"
- Non vi sono particolari pratiche da seguire

### Chi ha usato Scrum fino ad oggi:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit

- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

### Quando è stato usato Scrum:

- Software Commerciale
- Sviluppo interno
- Sviluppo a contratto
- Progetti a costo fisso
- Applicazioni finanziarie
- Applicazioni certificate (ISO 9001)
- Sistemi integrati (Embedded)
- Sistemi ad alta affidabilità (24x7 99.999% funzionamento)
- Avionica per caccia militare

- Sviluppo Video game
- Sistemi salvavita (US FDA)
- Sistemi controllo satelliti
- Siti Web
- Applicazioni Mobili
- Formware smartphone
- Sistemi di controllo del traffico Internet
- Applicazioni ISV
- Alcune delle più grandi applicazioni che oggi sono in uno...

## Il Manifesto Agile – una dichiarazione di valori

Individui e Interazioni

Software Funzionante

Collaborazione con il Cliente

Rispondere al Cambiamento

Processi e Tool

Documentazione Esaustiva

Negoziazione del contratto

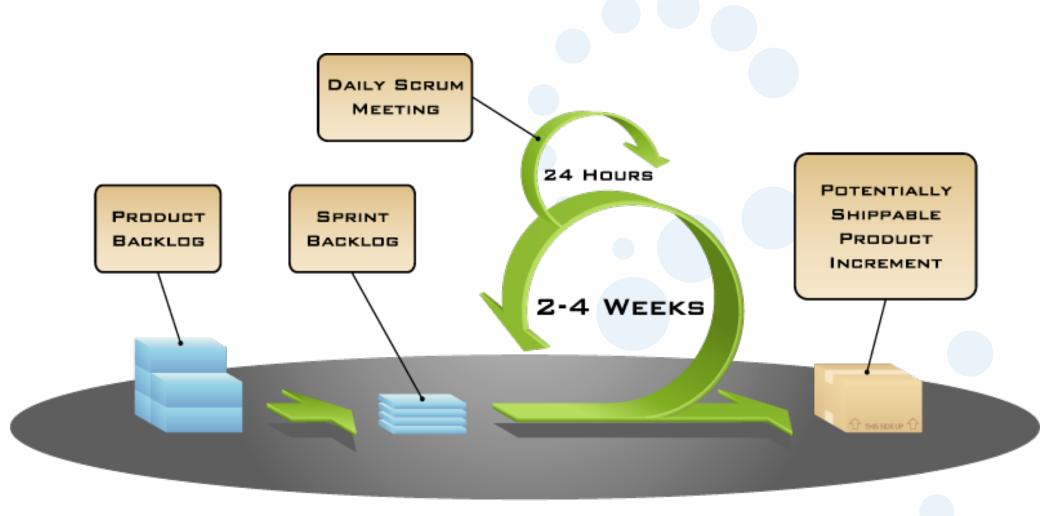
Seguire un Piano

Source: www.agilemanifesto.org

**Piuttosto** 

che

### Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Image available at www.mountaingoatsoftware.com/scrum

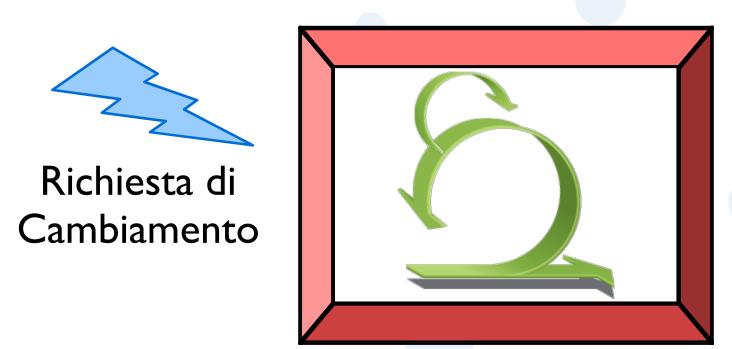
### Sprint

- Un progetto Scrum avanza attraverso una serie di "Sprints"
- Uno Sprint è simile ad un'interazione del modello a spirale
- La durata tipica di uno Sprint è 2-4 settimane o al massimo un mese di calendario
- Una durata costante porta ad avere un miglior ritmo
- Durante lo Sprint il prodotto viene progettato, realizzato e testato

Piuttosto che completare uno step alla volta, I team Scrum fanno sempre un po' di tutto ad ogni sprint

Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

## Nessun cambiamento durante lo Sprint



Stabilire e pianificare la durata dello sprint in accordo a quanto tempo è possibile "rifiutare" la richiesta di un cambiamento.

### II Framework Scrum

#### Ruoli

- Product owner
- ScrumMaster
- Team

#### Fasi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

#### Strumenti

- Product backlog
- Sprint backlog
- Burndown charts

### II Framework Scrum

#### Ruoli

- Product owner
- ScrumMaster
- Team

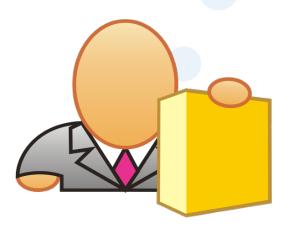
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

#### Strumenti

- Product backlog
- Sprint backlog
- Burndown charts

### Product Owner

- Definisce le funzionalità del prodotto
- Decide le date di rilascio e il contenuto
- È il responsabile della profittabilità (ROI) del prodotto
- Definisce la priorità tra le funzionalità in base al valore di mercato delle stesse
- Adatta le funzionalità dopo ogni sprint
- Accetta o rifiuta le funzionalità dopo ogni ciclo



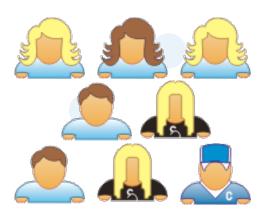
### ScrumMaster

- Rappresenta la figura manageriale del progetto
- È il responsabile dell'applicazione del framework Scrum, del suo miglioramento ed esecuzione delle pratiche
- Rimuove possibili impedimenti ed ostacoli
- Si assicura che il team sia pienamente funzionale ed operativo
- Facilita la collaborazione tra ruoli e funzioni
- "Protegge" il team dalle interferenze esterne



### **Team**

- Tipicamente composto da 5-9 persone
- Inter-funzionale:
  - Programmatori, tester, user experience designer, etc.
- I membri del team devono essere full-time
  - Alcune eccezioni possono essere fatte (es., amministratori del database)
- Il Team è auto-organizzato
  - Idealmente non viene assegnato alcun titolo, raramente è possibile
- I membri di un team sono gli stessi per la durata di tutto lo sprint, eventuali cambiamenti vengono effettuati solo tra uno sprint ed l'altro, mai durante!



### Scrum framework

#### Ruoli

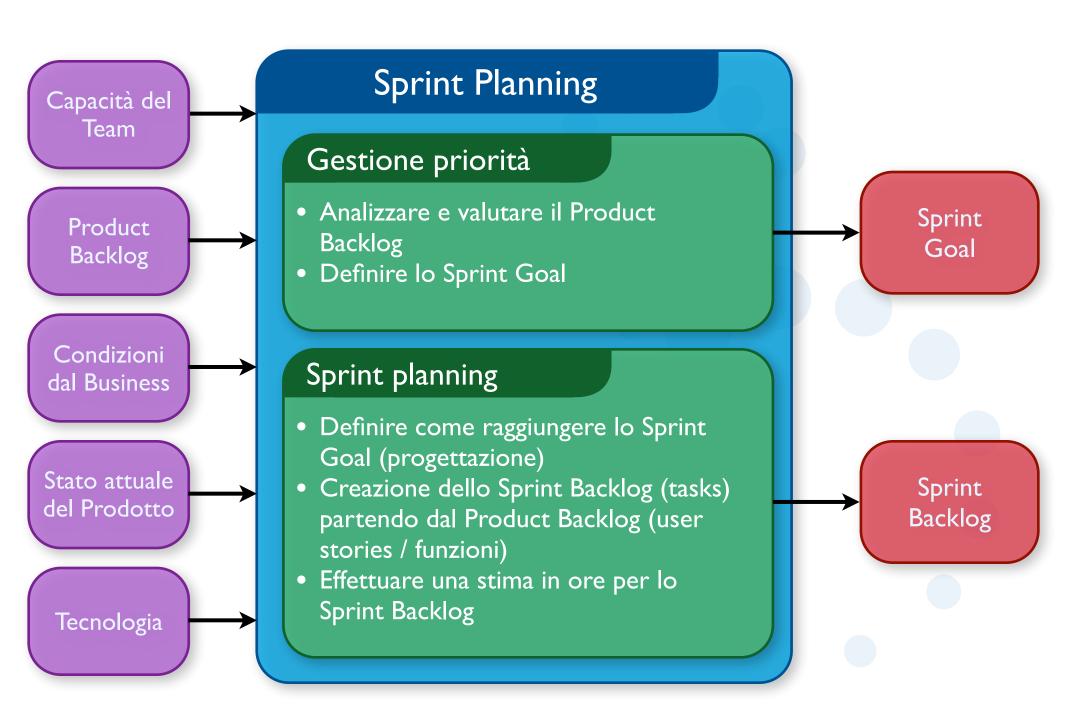
- Product owner
- ScrumMaster
- Team

#### Fasi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

#### **Strumenti**

- Product backlog
- Sprint backlog
- Burndown charts



### Capacità del Team

- Durante la pianificazione dello sprint, i team devono portare a termine i task dello sprint.
  - Quante stories si possono realizzare in questo sprint?
  - Come pianificare la capacità del team?
- La capacità del team è calcolabile tramite il numero di team member disponibili nello Sprint?
  - Esempio: Consideriamo un Team di 5 persone, possiamo assumere che con 8 ore di lavoro al giorno, in uno sprint di 2 settimane (10 giorni) tale valore sia = 5\*8\*10 = 400 ore? NOOOO!
  - Effettuare una pianificazione di questo tipo sarà un disastro, in quanto il team lavorerà per un tempo eccessivo, affrettandosi a finire il lavoro, la qualità degli artefatti sarà ridotta ed il morale del team sarà basso.

### Capacità del Team & Focus Factor

#### Focus Factor (F.F)

- È l'abilità del team di restare focalizzato sullo sprint goal senza soccombere ad altre distrazioni.
- Si aggira in un range tra 0.6 0.8
- Moltiplicando la capacità totale (ore totali di lavoro) con il focus factor è
  possibile ottenere la reale capacità del team, rispetto alla quale è possibile
  fare previsioni, in quanto essa rappresenta in maniera più adeguata le ore
  effettive di lavoro che ci si può aspettare dal team.
- Avere un focus factor di 0.6, significa che la reale capacità del team sarò di 400\*0.6 = 240 ore di lavoro.
- Il Team si occuperà delle stories fino a che tutti i componenti non raggiungeranno le 240 ore di lavoro (così come illustrato dal nostro esempio)

### Capacità del Team & Focus Factor

Usare un focus factor minore in situazioni come:

- Quando il team sta iniziando un nuovo progetto
- Quando il team usa Scrum per la prima volta
- Quando il team sta lavorando su un prodotto complesso o su un nuovo settore tecnologico
- Quando il team è immaturo e necessita di aiuti e supporto
- Quando le persone vengono assegnate a più progetti, si può verificare il problema del cambio di mansione.

Suggerimenti: Usare un fattore di focalizzazione minore quando si inizia da zero e poi, se il team raggiunge gli obiettivi dello sprint in anticipo, si può pensare di aumentarlo nello sprint in corso. Un buon indicatore può essere dato dalla Retrospective dello sprint, per comprendere se aumentare, diminuire o invariare il valore del focus factor. Inoltre, andare oltre lo 0,8 può essere rischioso e può far deragliare i team.

### Sprint Planning

- Il team seleziona gli elementi dal Product Backlog che può realizzare.
- Viene creato lo Sprint backlog
  - Vengono identificati i Tasks (User Story) e vengono stimati (I-16 ore) in modo collaborativo, tale processo non viene condotto dallo ScrumMaster
  - Viene considerata la progettazione ad alto livello

Come organizzatore di una vacanza, voglio vedere le foto degli hotel.

Codice Backend (8 ore)
Codice User Interface (4 ore)
Caso d'uso (6 ore)
Test Funzionale (4 ore)
Test Prestazioni (4 ore)

### **User Stories**

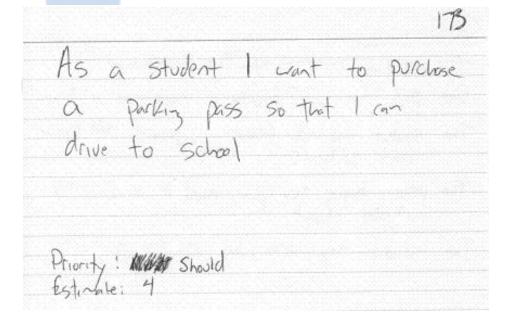
Artefatti di requisiti molto sottili e di alto livello (funzionali, non funzionali, vincoli...)

#### Formato:

Come ruolo lo voglio qualcosa in modo che beneficio

"In qualità di studente desidero acquistare un pass per il parcheggio in modo da poter andare a scuola in auto."

**User story card** (stakeholder)



### Stimare le User Stories

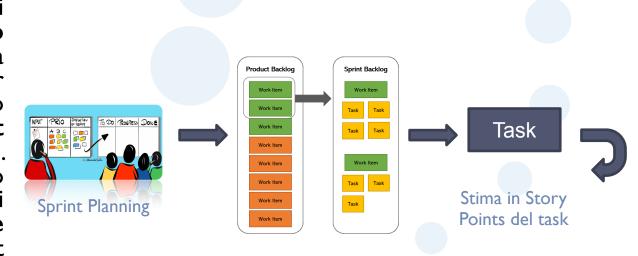


Il Poker planning in generale è un metodo utilizzato per stimare l'effort necessario per realizzare i requisiti di un prodotto. Noto anche come Scrum Poker, è principalmente utilizzato nell'ambito dello sviluppo software agile.

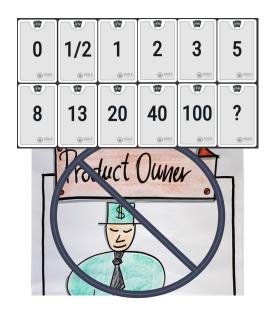
Punti Fondamentali del Poker Planning

- Il risultato della stima è espresso in Story Point
- La tecnica di Stima è basata sull'opinione di Esperti
- In Scrum, la stima è fatta nella fase di Sprint Planning

Gli Story Points sono un'unità di misura usata nel processo di sviluppo agile, per esprimere una stima dell'effort complessivo, necessario per implementare un'item, formalizzato nella specifica funzionale (Product Backlog, nel caso si usi Scrum). Essenzialmente, gli Story Points, sono utilizzati, al fine di far avere al team di progetto un'idea comune e standardizzata di quale sia l'effort necessario per i task da realizzare.



### Planning Poker



#### Step 1: Fase Individuale

Ogni membro del team suppone il numero di giorni in cui Secondo Lui il task si completa.

Si usa la successione di fibonacci:  $0, \frac{1}{2}, 1, 2, 3, 5, 8, 13, 20, 40, 100, (0 giorni, Mezza giornata, Un giorno, ...)$ 

#### Step 2: Fase Collettiva

Si mettono a confronto iterativamente le stime individuali, per tirarne fuori una comune (Come nel metodo Delphi)... La stima finale definisce il numero di Story Point del Task

#### Problemi del Planning Poker

- Stima altamente soggettiva;
- Valori che dipendono dal team;
- Non tenendo conto della dimensione dei requisiti non viene stonata la produttività.

### Daily Scrum Meeting

- Come farlo
  - Ogni giorno
  - 15 minuti
  - In piedi
- Non è il luogo in cui vengono risolti i problemi
  - Tutti possono partecipare
  - Solo i membri del team, lo ScrumMaster ed il product owner possono parlare
- Riduce le riunioni non necessarie

### Risponde a tre domande

Cosa hai fatto ieri?Cosa farai oggi?Qualche impedimento?

Tali domande non hanno l'obiettivo di coordinarsi con lo Scrum Master, ma di responsabilizzare tutti i membri del team e coordinare le loro attività.

### Sprint Review

- Il Team presenta al Product Owner cosa ha realizzato durante lo sprint
- Tipicamente tale presentazione viene condotta tramite demo delle funzionalità o dell'architettura dell'applicazione, che avviene in maniera informale
  - Solitamente dura 2 ore
  - È una demo, NO Slide
- Tutto il team partecipa allo sprint review
  - ScrumMaster
  - Product owner
  - Team
  - Eventualmente cliente ed altri



### Sprint Retrospective

- Periodicamente bisogna analizzare l'andamento dello sprint e cosa non ha funzionato
- Tipicamente dura 15–30 minuti
- Viene eseguito alla fine di ogni sprint
- L'intero team prende parte alla Sprint retrospective
  - ScrumMaster
  - Product owner
  - Team
  - Eventualmente cliente ed altri

### Start / Stop / Continue

Tutto il team riunito discute di cosa si preferirebbe fare

Cominciare a fare

Questo è solo uno dei tanti modi per eseguire una Retrospective Smettere di fare

Continuare a fare

### Scrum framework

#### Ruoli

- Product owner
- ScrumMaster
- Team

#### Fasi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

#### Strumenti

- Product backlog
- Sprint backlog
- Burndown charts

### Product Backlog

- Contiene la lista di tutti i requisiti
- Contiene la lista di tutte le funzionalità
- Idealmente costruito in modo che ogni elemento abbia valore per i clienti del prodotto
- Priorità gestita dal Product Owner
- Priorità verificata all'inizio di ogni Sprint

Product Backlog

### Esempio di Product Backlog

Backlog item	Stima	
Come ospite, voglio prenotare una camera	3	
Come ospite, voglio cancellare una precedente prenotazione	5	
Come ospite, voglio modificare le date di una precedente prenotazione	3	
Come impiegato dell'hotel, voglio eseguire un RevPAR reports (revenue-per-available-room)	8	
Migliorare la gestione delle eccezioni	8	
•••	30	
•••	50	

### Gestire lo Sprint Backlog

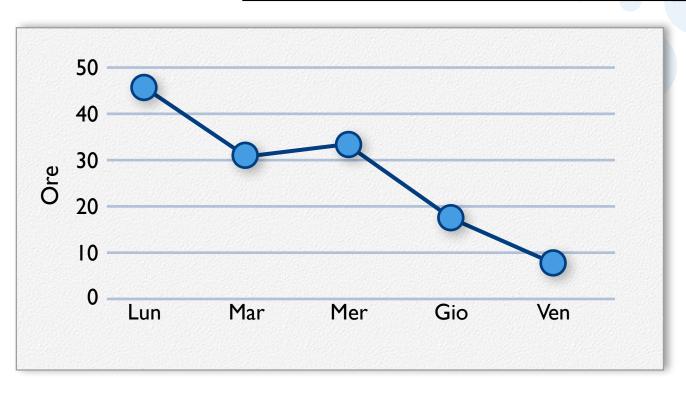
- Ognuno sceglie quali task realizzare
  - Il lavoro NON viene assegnato
- La stima del lavoro rimanente viene aggiornata quotidianamente.
- Ogni membro può migliorare i requisiti presenti nello Sprint Backlog
- Il lavoro sulle storie dello sprint "emerge"
- Se un requisito non è chiaro all'inizio dello Sprint, utilizzare il lavoro durante lo Sprint per andare a chiarirlo.
- Aggiornare il Product Backlog man mano che i requisiti risultano essere più chiari

### Esempio di Sprint Backlog

Tasks	Lun	Mar	Mer	Gio	Ven
Realizzare frontend	8	4	8		
Realizzare backend	16	12	10	4	
Testare il backend	8	16	16	11	8
Scrivere online help	12				
Scrivere classi d'esempio	8	8	8	8	8
Tracciare errori sul log			8	4	

### Esempio Sprint Burndown Chart

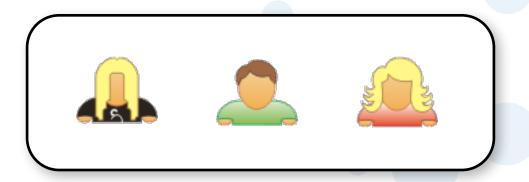
Tasks	Lun	Mar	Wed	Thur	Fri
Realizzare frontend	8	4	8		
Realizzare il backend	16	12	10	7	
Testare il backend	8	16	16	П	8
Scrivere l'online help	12				

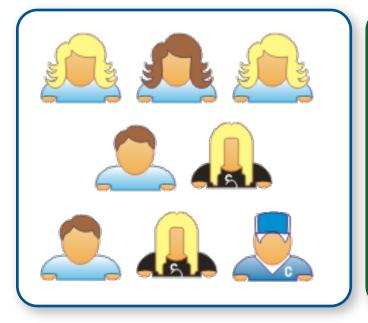


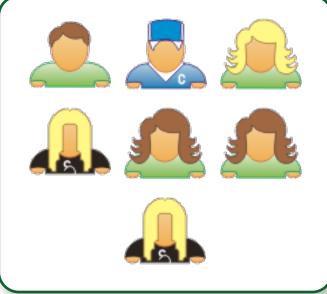
### Scalabilità di Scrum

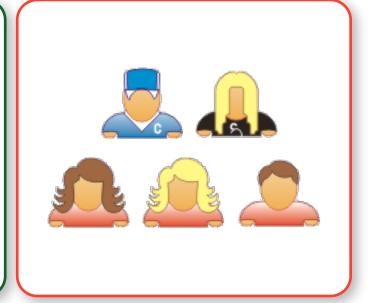
- Tipicamente la dimensione del team è 7 (± 2) persone
  - La scalabilità è gestita come "team di team"
- I fattori collegati alla scalabilità sono
  - Tipo di applicazione
  - Dimensione del team
  - Distribuzione del team
  - Durata del progetto
- Scrum è già stato utilizzato per progetti con 500+ persone (definiti Scrum of Scrums)

### Scalabilità con Scrum of Scrums









### Scrum of Scrums of Scrums





























### Libri Suggeriti





Presentation by: Mike Cohn mike@mountaingoatsoftware.com www.mountaingoatsoftware.com (720) 890-6110

### **SCRUMIA**

- Utilizzo della Gamification con l'intento di apprende l'utilizzo di Scrum.
- Uno scenario che fornisce ai giocatori le informazioni del progetto da realizzare (contesto, user story, durata degli sprint, ...)
- I partecipanti si dividono in gruppi:
  - Ogni giocatore ricopre un ruolo;
  - Ognuno di loro porta a termine le proprie loro mansioni.
  - Il primo gruppo di giocatori che porta a termine il progetto con successo vince.

Source: "SCRUMIA—An educational game for teaching SCRUM in computing courses" - The Journal of Systems and Software 2013.

The Journal of Systems and Software 86 (2013) 2675-2687



#### Contents lists available at ScienceDirect

#### The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss



#### SCRUMIA—An educational game for teaching SCRUM in computing courses

Christiane Gresse von Wangenheim\*, Rafael Savi, Adriano Ferreti Borgatto

Department of Informatics and Statistics, Federal University of Santa Catarina, Brazil

#### ARTICLE INFO

Article history:
Received 16 October 2012
Received in revised form 5 February 2013
Accepted 6 May 2013
Available online 14 May 2013

Keywords: SCRUM Teaching Game

#### ABSTRACT

Due to the increasing use of agile methods, teaching SCRUM as an agile project management methodology has become more and more important. In order to teach students to be able to apply SCRUM in concrete situations, often educational (simulation) games are used. However, most of these games have been developed more for professional trainings than taking into consideration typical restrictions of university courses (such as, class duration and low financial resources for instructional materials). Therefore, we present a manual paper and pencil game to reinforce and teach the application of SCRUM in undergraduate computing programmes complementing theoretical lectures. The game has been developed following a systematic instructional design process and based on our teaching experience. It has been applied several times in two undergraduate project management courses. We evaluated motivation, user experience and the game's contribution to learning through case studies on Kirkpatrick's level one based on the perception of the students. First results indicate the potential of the game to contribute to the learning of SCRUM in an engaging way, keeping students immersed in the learning task. In this regard, the game offers a low-budget alternative to complement traditional instructional strategies for teaching SCRUM in

© 2013 Elsevier Inc. All rights reserved.

#### 1. Introduction

The adoption of SCRUM as an agile project management methodology in software projects is constantly increasing (Abrahamsson et al., 2002). As a consequence, the need for software professionals with an understanding of SCRUM and how to apply it has never been greater than today (ACATE, 2011). And, therefore, these competencies have to become an essential part of the education of future software professionals (Layman et al., 2008). Yet, typically, in computing courses, project management (PM) (including SCRUM as an agile PM methodology) is still considered a minor topic (ACM/AIS/IEEE-CS, 2005), taught as only one of several topics in software engineering courses. As a result, students often do not feel confident with respect to their competency on PM and SCRUM (Gkritsi, 2011; Lethbridge, 2000). Many employers also find that recent graduates are not equipped with the desired PM competencies (Bennatan, 2011). Even though they may have theoretical PM knowledge, they certainly lack practical experience on its application. This indicates that PM education at universities commonly

0164-1212/\$ – see front matter © 2013 Elsevier Inc. All rights reserved http://dx.doi.org/10.1016/j.jss.2013.05.030 focuses only on theoretical aspects and does not give students the opportunity to apply PM to practice situations (Gkritsi, 2011).

An alternative to traditional lectures are educational games, which can provide various advantages as an instructional strategy (Percival et al., 1993). They can help to strengthen comprehension and to achieve deep learning within acceptable teaching time and instructor load (Wangenheim and Shull, 2009). And, due to their inherent characteristics, such as, competition, challenge and interaction, they can turn learning into an engaging experience having fun (Kafai, 2001).

For teaching SCRUM in the context of professional training courses, there exist a vast variety of hands-on activities and games that can be found on the Internet (mainly via blogs), such as, Tasty-Cupcakes (http://tastycupcakes.org). But, although in some cases even the complete game material is made available, most of these games lack a clear specification on how they are incorporated within an instructional context following a systematic instructional design. Furthermore, literature reporting their application also in university courses is basically inexistent. And, although games are believed to result in a wide range of benefits, like increased learning effectiveness, interest and motivation as well as a reduction of teaching time and instructor load (Jacobs and Dempsey, 1993; Prensky, 2001), these claims are arguable or at least not rigorously established (Akili, 2006) due to a lack of empirical studies evaluating systematically the impact of such games (Wangenheim and Shull, 2009).

<sup>\*</sup> Corresponding author at: Department of Informatics and Statistics – INE, Federal University of Santa Catarina – UFSC, 88049-200 Florianópolis, SC, Brazil. Tel.: +55 48 3721 9516; fax: +55 48 3331 9516xR16.

 $<sup>\</sup>label{lem:email} \textit{E-mail addresses}: \texttt{gresse@inf.ufsc.br}, \texttt{gresse@gmail.com} (C.G. von Wangenheim), \\ \texttt{rafaelsavi@gmail.com} (R. Savi), \texttt{borgatto@inf.ufsc.br} (A.F. Borgatto).$